

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gašper Mis

**PAMETNA REGULACIJA OGREVANJA  
SANITARNE VODE S SONČNIMI KOLEKTORJI**

DIPLOMSKO DELO  
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Ljubljana, 2016



UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gašper Mis

**PAMETNA REGULACIJA OGREVANJA  
SANITARNE VODE S SONČNIMI KOLEKTORJI**

DIPLOMSKO DELO  
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: viš. pred. dr. Robert Rozman

Ljubljana, 2016





To diplomsko delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* ali (po želji) novejšo različico. To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, dajejo v najem, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani <http://creativecommons.si/> ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njenih rezultatov in v ta namen razvite programske opreme je ponujena pod GNU General Public License, različica 3 ali (po želji) novejšo različico. To pomeni, da se lahko prosto uporablja, distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Izdelajte zasnovano sistema za učinkovito regulacijo ogrevanja sanitarne vode v stanovanjski hiši s kombinacijo dveh energetskih virov: sončnih kolektorjev in električnega grelca.

Sistem tudi realizirajte z vso potrebno strojno in programsko opremo. Proces ogrevanja sanitarne vode poskusite optimizirati, predvsem z vidika stroškov porabljene električne energije. Pri tem raziščite možnosti uporabe napovedi vremenskih razmer (predvsem osončenosti) in dvotarifnega merjenja porabe električne energije.

S pomočjo simulacij in analize delovanja sistema v testnem okolju oblikujte splošna priporočila za učinkovito ogrevanje sanitarne vode v tovrstnih objektih.



# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani      Gašper Mis,

z vpisno številko      63000221,

sem avtor diplomskega dela z naslovom:

Pametna regulacija ogrevanja sanitarne vode s sončnimi kolektorji

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom  
viš. pred. dr. Robert Rozman
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek  
(slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko  
diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki  
"Dela FRI".

V Ljubljani, dne 30.8.2016

Podpis avtorja:



Zahvalil bi se mentorju za pomoč pri izdelavi diplomske naloge. Dr. Andreju Kosu za idejo in nasvete. Nevi Pristov iz Agencije Republike Slovenije za okolje za generiranje in pošiljanje vremenske napovedi. Svoji ženi, otrokom ter bližnjim za potrpežljivost in podporo.





*Svoji sestri Urški v spomin.*



# Kazalo

<b>Povzetek</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>1 Uvod</b>	<b>3</b>
<b>2 Opis testnega okolja</b>	<b>5</b>
<b>3 Zasnova regulatorja ogrevanja sanitarne vode</b>	<b>7</b>
<b>4 Strojna oprema regulatorja</b>	<b>9</b>
4.1 Kartični računalnik Raspberry Pi . . . . .	9
4.1.1 GPIO priključek . . . . .	11
4.2 Temperaturni senzor . . . . .	11
4.3 Krmiljenje električnega grelca . . . . .	13
4.3.1 Vezje . . . . .	13
4.3.2 Optosklopnik . . . . .	15
4.3.3 Tranzistor . . . . .	15
4.3.4 Rele . . . . .	16
4.3.5 Kontaktor . . . . .	16
4.4 Namestitev regulatorja . . . . .	17
<b>5 Programska oprema regulatorja</b>	<b>19</b>
5.1 Programska arhitektura sistema . . . . .	19
5.2 Uporabniški vmesnik . . . . .	20
5.3 Uporaba napovedi sončnega sevanja . . . . .	23
5.4 Zajemanje podatkov s temperaturnih senzorjev . . . . .	24
5.5 Vklapljanje električnega grelca . . . . .	26

<b>6</b>	<b>Optimizacija ogrevanja sanitarne vode</b>	<b>27</b>
6.1	Vrednotenje količine tople vode . . . . .	27
6.2	Samooohlajanje bojlerja . . . . .	31
6.3	Raba tople sanitarne vode . . . . .	32
6.4	Solarno ogrevanje bojlerja . . . . .	32
6.5	Ogrevanje bojlerja z električnim grelcem . . . . .	38
6.6	Algoritem optimizacije . . . . .	38
6.6.1	Algoritem solarne optimizacije . . . . .	39
6.6.2	Algoritem za zagotavljanje minimalne količine tople vode	41
<b>7</b>	<b>Delovanje sistema</b>	<b>43</b>
7.1	Simulacija A: obstoječe samooohlajanje bojlerja . . . . .	44
7.2	Simulacija B: zmanjšano samooohlajanje bojlerja . . . . .	45
7.3	Simulacija C: izboljšan regulacijski algoritem . . . . .	45
7.4	Ocena učinkovitosti izboljšanega regulacijskega algoritma . . . .	46
<b>8</b>	<b>Sklepne ugotovitve</b>	<b>49</b>
	<b>Seznam slik</b>	<b>49</b>
	<b>Seznam tabel</b>	<b>52</b>
	<b>Seznam primerov kode</b>	<b>53</b>
	<b>Literatura</b>	<b>54</b>

# Seznam uporabljenih kratic in simbolov

V diplomski nalogi so uporabljene sledeče kratice in simboli:

kratica	izraz	prevod
RAM	Random Access Memory	bralno pisalni pomnilnik
USB	Universal Serial Bus	univerzalno serijsko vodilo
SD	Secure Digital	tip pomnilniške kartice
GPIO	General Purpouse Input Output	splošen vhodno izhodni priključek
ROM	Read Only Memory	bralni pomnilnik
ARSO	Agencija Republike Slovenije za Okolje	
XML	Extensible Markup Language	standard oblike podatkov, primerne za izmenjavo med sistemi
IMAP	Internet Message Access Protocol	protokol za dostop do elektronske pošte



# Povzetek

V diplomskem delu so bile raziskane možnosti za optimizacijo ogrevanja sanitarne vode s pomočjo sončnih kolektorjev in električnega grelca. Ker je električna energija v nočnem času cenejša, bi bilo smiselno ugotoviti, ali se splača dogrevati bojler z električnim grelcem samo ponoči.

Sestavljen je bil pametni regulator, ki na podlagi vremenske napovedi in predvidene porabe tople vode, vklaplja grelec večinoma le v nočnem času in po potrebi. Strojna oprema skrbi za branje podatkov s temperaturnih senzorjev in za vklop električnega grelca. Programski del sistema pridobiva vremensko napoved iz elektronske pošte in na podlagi analize napovedi ter predvidene porabe tople vode vklopi grelec.

Narejena je bila analiza delovanja algoritma, v kateri je bila primerjana učinkovitost algoritma ob različnih možnih scenarijih poteka osončenosti kolektorjev. Učinkovitost posameznega algoritma je bila določena na podlagi stroška električne energije, potrebne za zagotavljanje tople sanitarne vode. Ugotovljeno je bilo, da se v danem primeru ogrevanje vode izključno v nočnem času ne splača, ker so toplotne izgube boilerja prevelike. Primerjana je bila učinkovitost algoritma ob predpostavki polovice manjših izgub oziroma boljši izolaciji boilerja. Ugotovljeno je bilo, da je algoritem bolj učinkovit. Preizkušen je bil tudi izboljšan algoritem, ki v nočnem času ogreje vodo samo za dopoldansko porabo, popoldan pa ogreva vodo po potrebi. Ta algoritem je bil v testnem okolju uporabljen kot končna rešitev, ker prinaša največje prihranke.

## Ključne besede:

optimizacija ogrevanja, vremenska napoved, Raspberry Pi, senzorji, vklop električnih naprav, sončna energija

# Abstract

The thesis explores options for the optimization of sanitary water heating using solar panels and electric heater. Since electricity is cheaper at night, it would be reasonable to determine whether the electric heater should only be additionally heated at night alone.

A smart controller was assembled, which starts the heater on the basis of weather forecasts and estimated consumption of hot water, thereby mostly only at night and if necessary during the day. Hardware is responsible for reading data from temperature sensors and starting the electric heater. The software part of the system obtains a weather forecast from e-mail and starts the heater on the basis of forecast analysis and anticipated use of hot water.

An analysis of control algorithm was performed, which compared the effectiveness of the algorithm in different daily scenarios of collectors' sun exposure. The effectiveness of each algorithm was assessed on the basis of the cost of electricity necessary for the provision of domestic hot water. It has been concluded that in this case, heating water during night time alone is not worthwhile because the heat loss of the heater is too significant. Efficacy of the algorithm was compared assuming smaller losses due to better isolation of the heater. It has been found that the algorithm is more efficient. An improved algorithm which heats water only for morning use and heats it during the day only when needed, was put to test as well. This algorithm was used as a final solution in the test environment, as it delivers the greatest savings.

## Key words:

optimization of heating, weather forecast, Raspberry Pi, sensors, electrical appliances control, solar energy



# Poglavje 1

## Uvod

Priprava tople sanitarne vode v povprečnem gospodinjstvu predstavlja približno 10% vseh energijskih potreb. Tako naj bi za ogrevanje sanitarne vode porabili med 1,2kWh in 2,4kWh energije na osebo na dan [30], kar ni zanemarljivo.

Na drugi strani pa imamo brezplačno sončno energijo v zadostnih količinah. Prav zato se vedno več gospodinjstev odloča za ogrevanje vode s sončnimi kolektorji. Tako naj bi imeli v Sloveniji nameščenih za približno 6m<sup>2</sup> sončnih kolektorjev na 1000 prebivalcev [7], število pa še narašča. V optimalnih pogojih lahko z njimi pokrivamo do 80 odstotkov potreb po sanitarni vodi [7]. Na žalost pogoji niso vedno idealni in sonce ne sije ves čas.

V obdobjih, ko ni dovolj sončnega sevanja, moramo vodo ogrevati iz drugih virov energije in mnogi za to uporabljamo električni grelec. Glede načina vklapljanja grelca pa smo omejeni na dve možnosti. Lahko uporabljamo preprosto toplotno stikalo (termostat), ki grelec vklopi, kadar temperatura v bojlerju pade pod določeno vrednost. Je najenostavnejša rešitev in ima najmanjšo porabo električne energije ob predpostavki fiksne cene električne energije (je pa res, da včasih ne zagotovi tople vode dovolj hitro). Problem pa je, da je večina porabe v dražji tarifi in je s tem cena višja. Druga možnost pa je, da z uporabo časovnika vodo ogrevamo ponoči, ko je cena elektrike nižja. Segrevati jo moramo na višjo temperaturo. To pa v primeru sončnega dneva ni smiselno, ker imamo potem višek tople vode in smo zaman porabili električno energijo.

Cilj te naloge je razviti bolj učinkovit regulator ogrevanja. Na podlagi vremenske napovedi bo ugotovil količino vode, ki jo bomo pridelali s kolektorji. To bo primerjal s predvideno porabo tople vode. Vodo bo ogreval samo v nočnih časih, ko je cena elektrike nižja. Segrel bo samo potrebno količino vode. Pričakujemo, da bomo s tem ukrepom znižali strošek ogrevanja vode.

Hkrati pa bomo porabo elektrike predstavili v čas, ko imamo v električnem omrežju viške električne energije in s tem razbremenili električno omrežje.

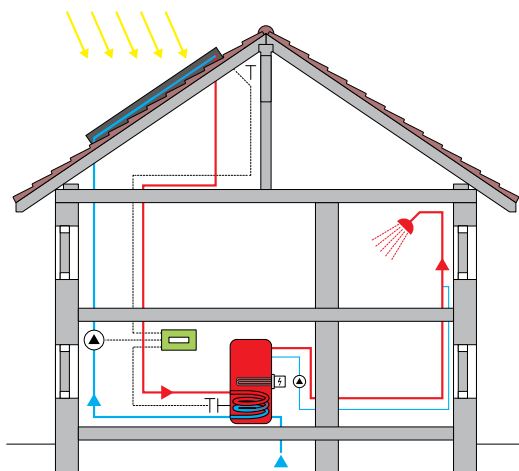
V nadaljevanju diplomske naloge je najprej opisano realno testno okolje za preizkus regulatorja ogrevanja sanitarne vode. Sistem se nahaja v stanovanjski hiši in vključuje uporabo sončnih kolektorjev in električnega grelca za ogrevanje sanitarne vode za petčlansko družino. Opišemo zasnovo regulatorja, v kateri podamo osnovne značilnosti samega regulatorja. Zatem je podrobneje opisana strojna oprema, potrebna vezava senzorjev in vklop/grelnje grelca. Nato opišemo programsko opremo in glavne komponente, kot sta branje podatkov s senzorjev in vklop grelca. Pri optimizaciji ogrevanja sanitarne vode smo se posvetili gradnji modelov s katerimi opišemo, kako vrednotimo količino tople vode, ovrednotimo porabo tople sanitarne vode, ter raziščemo vpliv sončnega sevanja na količino tople vode v bojlerju. Poglavje o delovanju sistema prikazuje razlike med sedanjim ogrevanjem po potrebi in izboljšano regulacijo solarnega ogrevanja. Prikazani so trije scenariji z različnimi parametri. Na koncu so na kratko povzete še možne izboljšave.

## Poglavje 2

### Opis testnega okolja

Optimizacijo ogrevanja sanitarne vode bomo testirali v lastni hiši, v kateri prebiva petčlanska družina. Nahaja se v okolici Šmarne gore. Stoji na odprtem in v bližini ni objektov, ki bi metali senco na streho. Izjema je sam hrib, ki do osme ure zjutraj senči celotno hišo. Hiša ima dvokapno streho, sleme pa je za 10 stopinj zamaknjeno od smeri vzhod-zahod.

Na sliki 2.1 je vidna trenutna shema ogrevanja. Hiši toplo vodo zagotavljata dva ploščata sončna kolektorja na južni strani strehe. Skupna površina sončnih kolektorjev je  $5\text{m}^2$ . Topla voda se zbira v bojlerju prostornine 300 litrov. V obdobjih, ko ni dovolj sončnega sevanja, pa toplo vodo zagotavlja



Slika 2.1: Shema trenutnega sistema ogrevanja sanitarne vode [2]

električni grelec v bojlerju. Vklaplja se preko termostata in se vklopi vsakič,

ko temperatura v bojlerju pade pod 40 °C. Električni grelec in termostat sta po višini vstavljena približno na polovici bojlerja. Električni grelec tako vedno segreva največ polovico bojlerja. V hiši je vgrajeno še kroženje tople vode. Poganja ga črpalka, ki se preko časovnika vklaplja ob vnaprej predvidenih časovnih intervalih zjutraj ob 6 uri za 1 uro, popoldne ob 16 za 1 uro, ter zvečer ob 19 za 1 uro.

Toplo vodo porablja petčlanska družina. Topla voda se porablja tekom celega dneva, z viški zjutraj in zvečer. V mesecih od sredine maja do sredine septembra sončna kolektorja segrejeta zadostno količino tople vode. Izjema so deževni dnevi. V jesenskem obdobju od sredine septembra do sredine novembra mora električni grelec stalno dogrevati toplo vodo. V tem obdobju je tudi največ dopoldanske megle, ki še povečuje uporabo električnega grelca. V spomladanskem okolju od sredine marca do sredine maja električni grelec prav tako stalno dogreva toplo vodo.

V zimskem času vodo segrevamo s kotlom na drva, s katerim ogrevamo tudi hišo. Ker segrevanje tople vode s kotlom ni tema oziroma del diplomske naloge, je izpuščen iz sheme na sliki 2.1 in se ga v samem delu ne omenja.

Glavna slabost trenutno uporabljenega sistema je, da vodo večinoma segreva v obdobjih, ko je cena električne energije višja.

## Poglavje 3

# Zasnova regulatorja ogrevanja sanitarne vode

V okviru diplomske naloge želimo sestaviti optimalni oziroma pametni regulator ogrevanja. Nadomestil bi trenutno precej enostavno rešitev ogrevanja po potrebi, ko se grelec vklopi, kadar temperatura v bojlerju pade pod določeni nivo. Ta rešitev je res enostavna, a stroškovno manj učinkovita.

Regulator naj ima sledeče lastnosti:

- Stroški strojne in programske opreme regulatorja naj skupno ne presegajo 100,00€.
- Delovati mora popolnoma samostojno, brez ročnega vnašanja parametrov.
- Regulator mora ugotoviti, če imamo v danem dnevu dovolj sončnega sevanja za segretje sanitarne vode. V nasprotnem primeru mora v nočnem času, ko je cena električne energije nižja, segreti vodo z električnim grelcem. Segreti mora samo potrebno količino vode (upoštevati mora napovedi porabe in količine vode, ki jo bo tekom dneva segrelo sonce).
- Imeti mora enostaven in pregleden vmesnik za analizo in nadzor delovanja.

S cenovno omejitvijo in z omejitvijo stroškov delovanja želimo povečati dostopnost regulatorja. Pri tem moramo biti skromni pri izbiri strojne opreme. Na srečo je na trgu dovolj tako imenovanih kartičnih računalnikov, ki imajo nizko ceno, hkrati pa tudi dovolj računske moči za naše potrebe. Odločili smo se za kartični računalnik Raspberry Pi, ker ga poznamo in je v primerjavi z

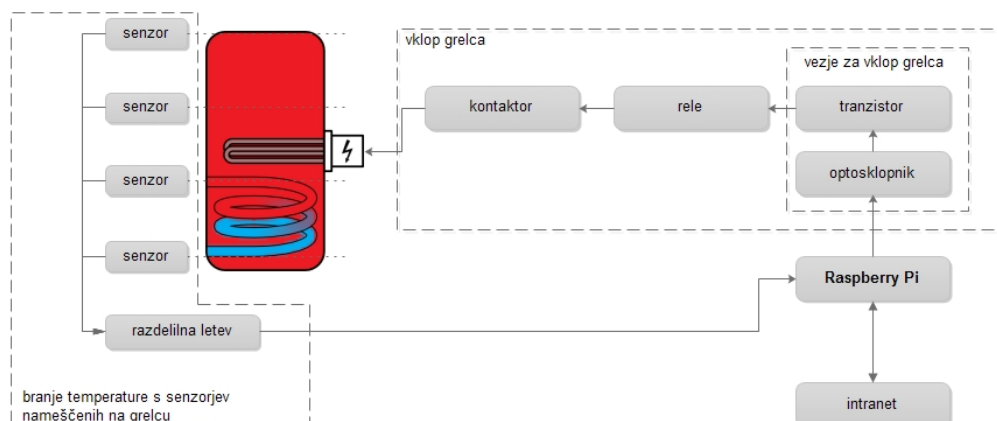
ostalimi rešitvami najbolj razširjen. Pri izbiri senzorjev in ostalih komponent pa smo gledali na to, da so lahko dostopni ter je njihova cena sprejemljiva.

Uporabljali smo samo odprtokodno (angl. “open source”) programsko opremo. Ugotavljamo, da je dovolj zanesljiva za uporabo, hkrati pa je brezplačna.

## Poglavje 4

# Strojna oprema regulatorja

V tem poglavju so opisane strojne komponente, potrebne za izdelavo regulatorja in njihove medsebojne povezave. Shema na sliki 4.1 prikazuje vezavo komponent strojne opreme. Najprej bomo opisali osrednjo komponento kartični računalnik Raspberry Pi. Nato bomo opisali vezavo temperaturnih senzorjev. Nazadnje pa še komponente potrebne za vklop grelca.



Slika 4.1: Shema povezav strojne opreme regulatorja

### 4.1 Kartični računalnik Raspberry Pi

Osrednji del sistema je kartični računalnik Raspberry Pi (slika 4.2), ki spada med najbolj uporabljene platforme za vgrajene naprave in naprave interneta stvari (angl. “Internet of Things” skrajšano “IoT”). Osnovan je na ARM procesorju [21], ki uporablja RISC arhitekturo. Ima odlično razmerje med računsko





S to izbiro smo dosegli, da bo en modul lahko izvajal vse funkcije sistema:

- krmiljenje grelca,
- branje podatkov s senzorjev,
- hranjenje podatkov,
- analizo podatkov,
- spletni vmesnik za nastavitve,
- pregled delovanja sistema.

### 4.1.1 GPIO priključek

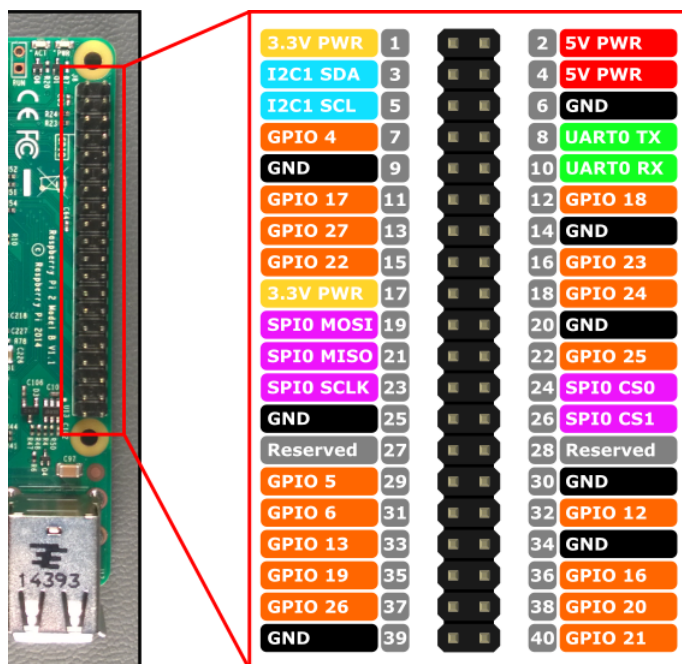
Kratika GPIO (“General Purpose Input Output”) dejansko pomeni splošno uporaben vhodni ali izhodni priključek. S tem poimenujemo 40 priključkov, ki se nahajajo ob robu kartičnega računalnika. Preko teh priključkov lahko kartični računalnik neposredno komunicira s senzorji ali ostalimi napravami. Priključki lahko imajo na izhodu in vhodu največjo napetost 3.3V. Izjema so le priključki, ki zagotavljajo 5V napetost (“5V PWR”) in ozemljitev (“GND”). Pri uporabi GPIO priključkov je treba biti še posebej pazljiv, kajti modul nima zaščite pred previsoko napetostjo. Skupni tok, ki lahko teče skozi izhode, je 50mA. Prevelika napetost ali tok pa lahko povzročita okvaro enote.

Priključki se potem še posebej poimenujejo glede na namen, kot je razvidno iz slike 4.3. Tu bi omenili samo še priključek GPIO, po katerem je sklop 40 priključkov dobil ime. GPIO priključek je lahko uporabljen kot vhodni ali izhodni, odvisno od nastavitve. Če deluje kot vhodni, lahko z njim preberemo ali imamo na priključku kontrolni tok. To je lahko stikalo, ki ga je pritisnil uporabnik ali katera druga komponenta. V nasprotnem primeru pa nam služi kot stikalo za vklop naprav. Z njim lahko s pomočjo dodatnega vezja vklapljammo tudi zelo visoke napetosti.

V diplomskem delu v primeru sklicevanja na GPIO priključek vedno mislimo na enega izmed 40 priključkov.

## 4.2 Temperaturni senzor

Za merjenje količine tople vode potrebujemo 4 temperaturne senzorje. Odločili smo se za senzorje z oznako DS18B20 (slika 4.4) proizvajalca Maxim Integrated [18]. Senzor meri temperaturo v območju od  $-55^{\circ}\text{C}$  do  $125^{\circ}\text{C}$ , ter ima



Slika 4.3: Raspberry Pi 2 seznam priključkov [26]

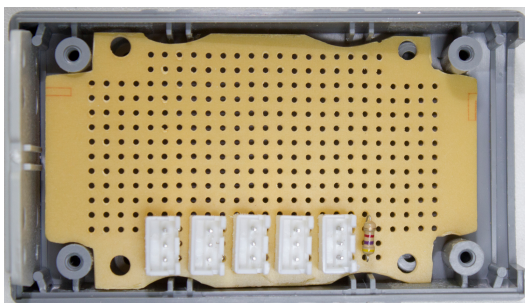
natančnost  $0.5^{\circ}\text{C}$  v razponu od  $-10^{\circ}\text{C}$  do  $85^{\circ}\text{C}$ . Napajamo ga z napetostjo od  $+3.0\text{V}$  do  $+5.5\text{V}$ , ter v povprečju rabi  $1\text{mA}$  toka. Z ostalimi napravami komunicira preko “1-Wire” vodila.



Slika 4.4: Temperaturni senzor DS18B20 [23]

Vsaka naprava na tem vodilu ima svojo unikatno 64bitno številko, kar omogoča priklop teoretično neomejenega števila naprav. Prednost je tudi v tem, da za delovanje lahko uporabimo samo dve žici. Vezavo poimenujemo tudi “parazitsko napajanje”. V tem načinu se podatkovna povezava uporablja tudi za napajanje. Vsak senzor vsebuje majhen kondenzator, ki se v času nedejavnosti napaja iz podatkovnega vodila. V času prenosa podatkov pa ima dovolj moči

za napajanje senzorja. Druga povezava pa je ozemljitev.



Slika 4.5: Razdelilna letev za temperaturne senzorje

V naši rešitvi smo uporabili 3 vodnike. Prvega kot napajanje, drugega kot ozemljitev in tretjega kot podatkovno vodilo. Uporabili smo tudi možnost namestitve več naprav na istem vodilu in sestavili razdelilno letev (slika 4.5), ki je nameščena na bojlerju. Prav tako smo na letev dodali še  $4.7\text{ k}\Omega$  “pull-up” upornik.

Na sam senzor smo prispajkali trižilni kabel. Kabel smo zaključili s konektorji s tremi nožicami. Na razdelilno letev smo priključili posamezne temperaturne senzorje, ki so preko skupnega vodila povezani na kartični računalnik. S samo razdelilno letvijo smo zmanjšali skupno dolžino kablov, ki so potrebni. Hkrati pa smo tudi dosegli boljšo preglednost, saj do kartičnega računalnika vodi samo en kabel s tremi vodniki.

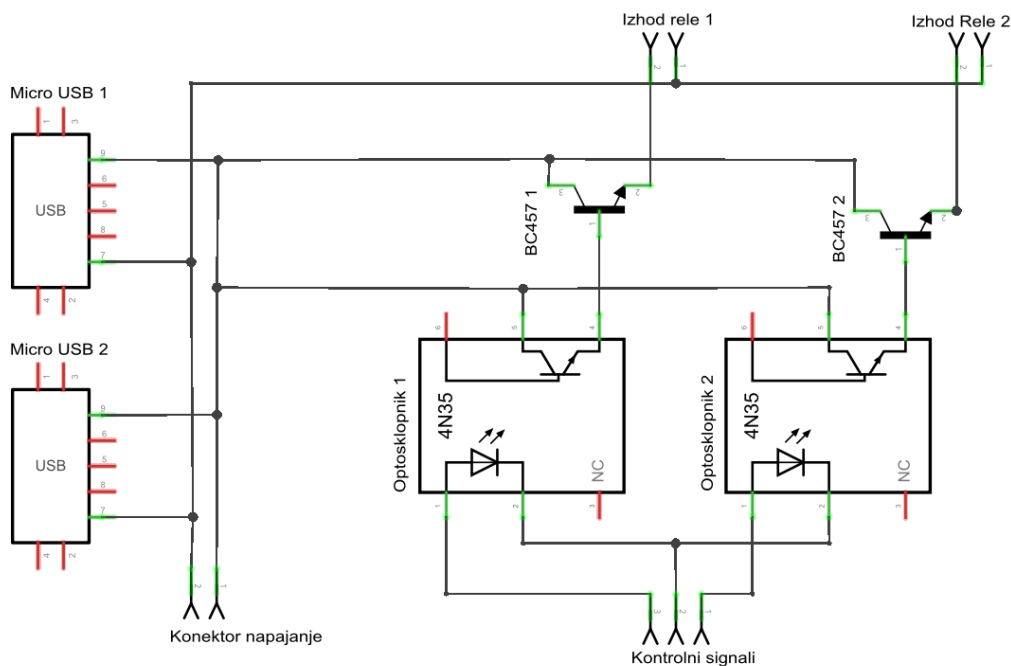
## 4.3 Krmiljenje električnega grelca

Za gretje sanitarne vode je uporabljen klasični spiralni električni grelec z močjo  $3\text{ kW}$ . Grelec deluje pri napetosti  $220\text{ V}$  in rabi  $16\text{ A}$  toka. Ker s kartičnim računalnikom ne moremo vklopiti tako močnih tokov, smo sestavili vezje z dodatnimi elementi, ki omogočajo vklop in izklop električnega grelca. V samem vezju smo uporabili več elementov, ki so vezani zaporedno. Z vsakim naslednjim v vezavi se preklaplja močnejši tok.

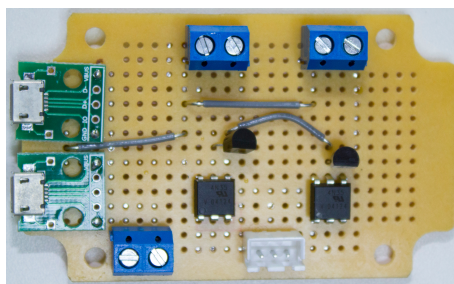
### 4.3.1 Vezje

Sestavili smo preprosto vezje, ki zagotavlja enostaven vklop električnega grelca (shema na sliki 4.6, slika 4.7). Vezje omogoča vklop dveh relejev istočasno. Napaja se iz navadnega polnilca za mobilni telefon. Polnilec zagotavlja napetost

5V in maksimalni tok 2A, kar je več kot dovolj za delovanje celotnega sistema. Priklop na polnilec je zagotovljen preko USB Micro B konektorja. Z istim polnilcem napajamo tudi Raspberry Pi, ki je na vezje priklopljen preko drugega USB Micro B konektorja. Dodali smo še drugi dvožilni konektor, ki služi za rezervo. Kontrolni signal za vklop relejev dovajamo preko trižilnega konektorja, pri čemer sta dve žili namenjeni za krmiljenje, srednja žila pa je negativni pol.



Slika 4.6: Shema vezja za vklop grelca



Slika 4.7: Izvedba vezja za vklop grelca

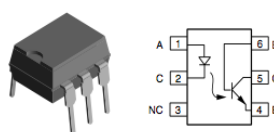
Za vklop grelca je potrebno zaporedno vezati elemente:

- optosklopnik 4N35, ki ga krmili kartični računalnik in vklaplja napetost 5V,
- tranzistor BC547, ki ga krmili optosklopnik in vklaplja napetost 5V, vendar z večjim tokom,
- rele JW2SN-5V, ki ga krmili tranzistor in vklaplja napetost 220V, vendar še ne more zagotavljati dovolj toka za vklop grelca,
- kontaktor KNL16, ki ga krmili rele in zagotavlja dovolj toka za vklop in delovanje grelca.

### 4.3.2 Optosklopnik

Optosklopnik je optično elektronsko stikalo, ki omogoča vklop večjih bremen z manjšimi. Razlika med optosklopnikom in relejem pa je v tem, da optosklopnik za vklop uporablja svetlobo. Na vhodnem delu vezja krmilni tok najprej aktivira svetlečo diodo, ki posveti na fototranzistor. Ta v tem primeru prične prevajati tok na izhodni strani optosklopnika.

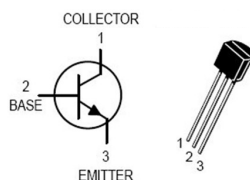
Glavna prednost optosklopnikov je v tem, da je med krmilnim tokom in krmiljenim tokom popolna izolacija in ne prihaja do elektromagnetnih motenj ob vklopu ali izklopu. Zato ne potrebujemo dodatne diode, ki bi preprečevala povraten sunek električnega toka ob izklopu. Uporabili smo optosklopnik 4N35 (slika 4.8) podjetja Texas Instruments [17]. Krmilimo ga z napetostjo 3.3V neposredno iz kartičnega računalnika.



Slika 4.8: Optosklopnik 4N35 [4]

### 4.3.3 Tranzistor

Tranzistor je polprevodniški element, ki se uporablja kot stikalo ali kot ojačevalec elektronskih signalov. Izbrali smo tranzistor BC547 [8] (slika 4.9), ki deluje pri napetosti 5V. Krmilimo ga preko optosklopnika in s tem kontroliramo delovanje releja.



Slika 4.9: Tranzistor BC547 [1]

#### 4.3.4 Rele

Rele je elektromehansko stikalo, ki omogoča krmiljenje višjih napetosti in tokov z nižjimi. Z releji običajno lahko sklapljamo napetosti do 220V in tokom do 5A. Običajno jih krmilimo z napetostjo od 5V do 24V. Osnova releja je tuljava. Ko skozi steče električni tok, se ustvari magnetno polje, ki pritegne kvoto k jedru. Premik kvote pa preklopi tudi stikalo, ki se vklopi. Ko kontrolni tok prekinemo, se izklopi tudi stikalo. Uporabili smo rele JW2SN-5V podjetja Panasonic [22], prikazan na sliki 4.10.



Slika 4.10: Rele JW2SN [9]

#### 4.3.5 Kontaktor

Kontaktor ali močnostni rele je elektromehanska komponenta, ki omogoča vklop močnih tokov. Deluje podobno kot rele, le da preklaplja še močnejše tokove. Prav tako pa za krmiljenje potrebuje večji tok in višjo napetost.

Uporabili smo kontaktor IKA20-20 [19] (slika 4.11) domačega proizvajalca Iskra, ki omogoča krmiljenje 20 A toka pri 220 V napetosti. Za vklop potrebuje napetost 220 V in v stanju sklopljenih kontaktov porabi 1.7 W energije. Krmilimo ga z električnim tokom, ki ga dovaja izhod releja.



Slika 4.11: Kontaktor IKA20-20 [27]

## 4.4 Namestitev regulatorja

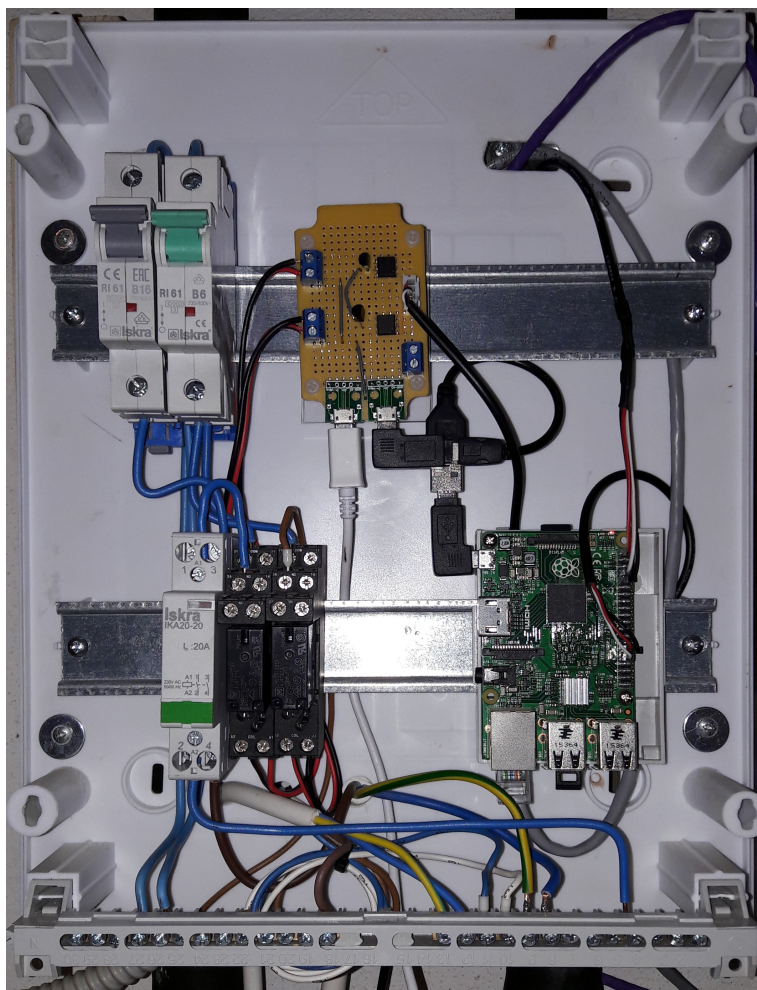
Vse komponente regularja smo namestili v plastično omarico, kot je prikazano na sliki 4.12. Spodaj desno je nameščen Raspberry Pi, ki je na intranet povezan preko mrežnega kabla. Ob desnem robu kartičnega računalnika se vidita priklopljena dva kabla (s tremi vodniki) na GPIO priključke. Zgornji kabel je povezan s temperaturnimi senzorji. Spodnji kabel pa je povezan na vezje za vklop grelca.

Zgoraj na sredini je vezje za vklop električnega grelca. Napaja se preko belega USB kabla na spodnji strani (za napajalnik služi polnilec za telefon). USB povezava med kartičnim računalnikom in vezjem za vklop grelca služi izključno samo napajanju kartičnega računalnika.

Spodaj levo je najprej nameščen kontaktor, ki ga krmili rele. Rele se nahaja na desni strani kontaktorja.

Na samem regulatorju sta nameščeni še dve varovalki (zgoraj levo). Služita za dodatno varnost pri delovanju. Opazimo tudi, da imamo na regulatorju še dodatni rele. Služi vklopu črpalke za kroženje vode in ni tema oziroma del te diplomske naloge.

Trenutna sestava regulatorja zavzema veliko prostora. Menimo, da bi lahko trenutno velikost regulatorja v morebitnem nadaljnjem razvoju zmanjšali za 60%.



Slika 4.12: Sestavljen regulator ogrevanja



## Poglavje 5

# Programska oprema regulatorja

V tem poglavju so opisane nameščene in implementirane programske komponente. Na kratko opišemo uporabniški vmesnik. Podrobneje si ogledamo pridobivanje in shranjevanje vremenske napovedi, funkcijo za branje vrednosti s temperaturnih senzorjev in funkcionalnost za vklopjanje električnega grelca.

### 5.1 Programska arhitektura sistema

V središču sistema je Raspberry Pi, na katerem teče celotna programska rešitev:

- branje in analiza podatkov s senzorjev,
- pridobivanje vremenske napovedi,
- vklop grelca,
- spletni vmesnik, kjer so na voljo informacije o stanju sistema.

Na sliki 5.1 je prikazana shema programske opreme. Za operacijski sistem smo izbrali Ubuntu mate Linux [16], stabilen in varen sistem. Odlikuje ga nizka poraba sistemskih sredstev (RAM, procesor, prostor na disku) in enostavna namestitvev programske opreme.

Za shranjevanje podatkov uporabljamo podatkovno bazo MySQL [5]. Spada med najbolj uporabljane podatkovne baze. Je enostavna za namestitvev in konfiguracijo. V podatkovno bazo shranjujemo podatke o vremenski napovedi in količini tople vode v bojlerju.

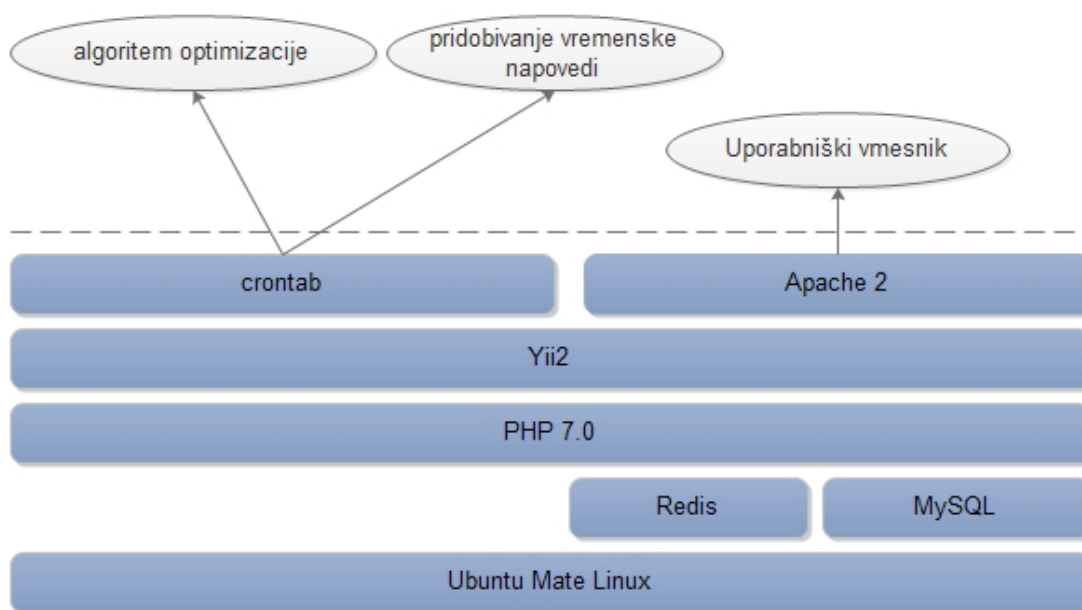
Za medpomnjenje (angl. “cache”) služi Redis strežnik [25]. Podatke začasno shranjuje v RAM, ki je mnogo hitrejši od branja iz datotečne strukture na SD kartici ali branja podatkov iz podatkovne baze. Dodatno shranjuje podatke o

prijavi uporabnika in s tem zmanjša število zapisov na SD kartico. Vse to pa pohitri delovanje uporabniškega vmesnika. Izbrali smo ga na podlagi preteklih izkušenj.

Celoten program je spisan v programskem jeziku PHP 7 [13], ki ga poznamo iz preteklih projektov. Uporabili smo ogrodje (angl. “framework”) Yii2 [20]. Ogrodje je sklop najbolj uporabljenih knjižnic in funkcij, ki pripomorejo k hitrejšemu razvoju programa. Yii 2 smo izbrali zaradi velike hitrosti delovanja in po našem mnenju dobro dokumentirane kode.

Uporabniški vmesnik teče na spletnem strežniku Apache2 [10].

Za periodično izvajanje operacij (pridobivanje vremenske napovedi, algoritem optimizacije) na samem sistemu pa skrbi “crontab”. Storitve (angl. “service”) je vključena v operacijski sistem in se enostavno konfigurira preko urejanja datoteke “/etc/crontab”.

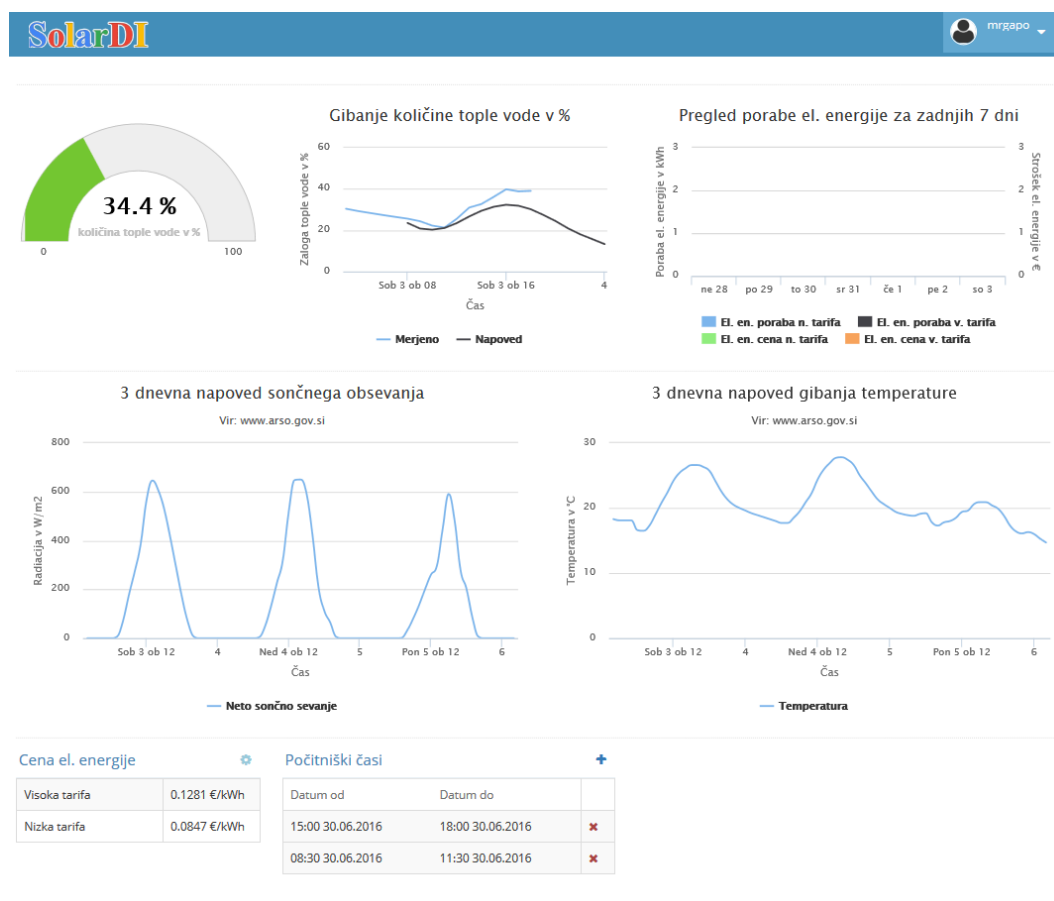


Slika 5.1: Shema programske opreme

## 5.2 Uporabniški vmesnik

Za nadzor in pregled delovanja sistema smo sestavili preprost uporabniški vmesnik (slika 5.2). Uporabili smo spletno predlogo, ki temelji na Twitter Bootstrap ogrodju (“framework”) [15]. Ta standardizira elemente strani tako,

da so lepo pregledni na vseh velikostih zaslonov. Kot osnovno ogrodje za postavitev spletne strani smo uporabili Yii2. Za dostop do vmesnika je potrebna prijava z uporabniškim imenom in geslom. Za prikaz grafov smo uporabili komponento Highcharts [14]



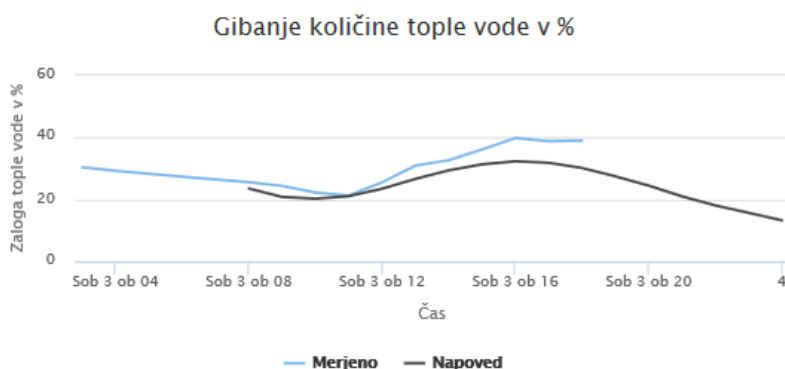
Slika 5.2: Uporabniški vmesnik

Uporabniški vmesnik na sliki 5.2 prikazuje naslednje informacije (od leve proti desni) :

- trenutno stanje količine tople vode,
- prikaz merjene in napovedane količine tople vode bojlerja za tekoči dan (slika 5.3),
- prikaz porabe električne energije za zadnjih 7 dni,

- graf tridnevne napovedi sončnega obsevanja,
- graf tridnevne napovedi gibanja temperature,
- tabela cen električne energije,
- tabela časov odsotnosti.

Graf gibanja količine tople vode tekom dneva na sliki 5.3, z modro črto prikazuje gibanje količine tople vode tekom dneva. S črno črto je prikazano napovedano gibanje količine tople vode, kot jo je predvidel algoritem optimizacije. Omogoča enostavno primerjavo napovedi gibanja količine tople vode našega algoritma z dejanskim stanjem. Ker smo sliko zajeli sredi dneva (po 18 uri), modra črta prikazuje stanje do te ure.



Slika 5.3: Graf gibanja količine tople vode tekom dneva

Graf porabe električne energije (slika 5.2 desno zgoraj) je prazen, ker v podanem obdobju ni bilo nobene porabe električne energije.

Tabela časov odsotnosti prikazuje obdobja, ko v hiši ni nikogar. Vmesnik omogoča brisanje in dodajanje novih počitniških časov preko vnosnega obraza.

Vnos je pomemben predvsem zaradi:

- algoritma za optimizacijo, ki ne prižiga grelca v času, ko v hiši ni nikogar,
- časi odsotnosti označujejo obdobje, ko ni porabe tople vode in omogočajo analizo vpliva sončnega sevanja na polnjenje bojlerja.

## 5.3 Uporaba napovedi sončnega sevanja

Za napoved sončnega obsevanja uporabljamo vremensko napoved ARSO. Pripravili so prilagojeno vremensko napoved, ki se izračunava za lokacijo, na kateri se nahaja hiša. Vremensko napoved pošiljajo vsak dan med poltretjo in tretjo uro zjutraj preko elektronske pošte. Napoved je pripeta v elektronskem pismu kot datoteka v XML obliki (primer kode 5.1).

Uporabljamo sledeče podatke iz XML datoteke:

- “<valid>” ura, za katero velja napoved,
- “<t>” temperatura v stopinjah Celzija,
- “<netSunRad\_1h\_acc>” neto sončno sevanje pri tleh (povprečni energijski tok v pretekli uri v  $\text{W/m}^2$ ).

Napisali smo podprogram za branje pošte in vnašanje podatkov. Poveže se z elektronskim predalom in pregleda, ali obstaja kakšno neprebrano elektronsko sporočilo. Za vsako neprebrano sporočilo pregleda ali vsebuje priponko. Če ustreza tipu XML in v imenu vsebuje niz `_SP_PIRNICE`, program sklepa, da je to vremenska napoved. Nato priponko shrani kot datoteko na trdi disk. Zatam pa podatke zapiše še v podatkovno bazo.

Podatke v bazo shranjujemo v tako imenovani ploščati tabeli. Vsi podatki za isto časovno obdobje se nahajajo v eni vrstici. V taki obliki so podatki shranjeni pregledno. Prav tako je takšna oblika primerna tudi za kasnejšo analizo. Slaba lastnost ploščatih tabel pa je v njihovi neprilagodljivosti. Vsakič, ko se doda nov tip podatka, je za to potrebno dodati polje v podatkovni bazi. Treba pa je popraviti tudi funkcije, ki zapisujejo ali pišejo v podatkovno bazo. Za to obliko smo se odločili, ker se sam podatkovni model ne bo spreminjal in imamo tako enostaven pregled podatkov, hkrati pa za potrebe optimizacije potrebujemo podatke v obliki, primerni za nadaljnje analiziranje.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2   <data id="MeteoSI_WebMet_forecast_xml">
3     <language>sl</language>
4     <metData>
5       <title>METEO_MODEL</title>
6       <domain_wmoId>1443.81,4613.45</domain_wmoId>
7       <domain_longTitle>c1443.81,4613.45</domain_longTitle>
8       <domain_meteoId>SP-PIRNICE</domain_meteoId>
9       <dataSource_title>ALEA</dataSource_title>
10      <domain_lat>46.1345</domain_lat>
11      <domain_lon>14.4381</domain_lon>
12      <domain_altitude/>
13      <modelOrographyHeight>341</modelOrographyHeight>
14      <tsValid>12.05.2016 0:00 UTC</tsValid>
15      <tsUpdated>12.05.2016 2:53 UTC</tsUpdated>
16      <valid>12.05.2016 2:00 CEST</valid>
17      <valid_UTC>12.05.2016 0:00 UTC</valid_UTC>
18      <leadTime>0:00</leadTime>
19      <t>12.1</t>
20      <gSunRad_1h_acc/>
21      <ff_val>0.8</ff_val>
22      <nn_val>7.5</nn_val>
23      <netSunRad_1h_acc/>
24    </metData>

```

Primer kode 5.1: Primer vremenske napovedi v XML obliki

## 5.4 Zajemanje podatkov s temperaturnih senzorjev

Podatke zajemamo s senzorjev s pomočjo knjižnice w1-gpio. Knjižnico je treba najprej aktivirati z zapisom v konfiguracijsko datoteko. Na konec datoteke “/boot/config.txt” se doda zapis “dtoverlay=w1-gpio”. Po ponovnem zagonu sistema izvedemo še ukaza “sudo modprobe w1-gpio” in “sudo modprobe w1-therm”. Posamezni temperaturni senzorji se vidijo kot mape na disku, pri čemer je ime mape id senzorja (slika 5.4). Vrednosti posameznega senzorja

```

root@raspberrypi:/srv/diploma# ls /sys/bus/w1/devices/
28-0000004f72be9 28-00000051f66ab 28-00000051f7ebc 28-00000051fb223 w1_bus_master1
root@raspberrypi:/srv/diploma#

```

Slika 5.4: Prikaz temperaturnih senzorjev v operacijskem sistemu

pa dobimo tako, da preberemo datoteko “w1-slave”, ki se nahaja znotraj di-

rektorija (slika 5.5). Temperatura je podana v drugi vrstici v obliki besedila

```
root@raspberrypi:/srv/diploma# cat /sys/bus/w1/devices/28-000004f72be9/w1_slave
db 03 4b 46 7f ff 05 10 5d : crc=5d YES
db 03 4b 46 7f ff 05 10 5d t=61687
root@raspberrypi:/srv/diploma#
```

Slika 5.5: Prikaz zapisa vrednosti temperaturnega senzorja

in se začne z nizom "t=<temperatura>". Podana je v stopinjah Celzija brez decimalne vejice z natančnostjo treh decimalk.

Za branje vrednosti senzorjev smo napisali funkcijo (primer kode 5.2), ki poišče obstoječe temperaturne senzorje in za vsakega prebere njegovo vrednost. Rezultat vrne v obliki niza objektov tipa `sensor`.

```
1 class oneWire{
2     public static function readSensorsValue(){
3
4         $sensorsValue = [];
5         $handle = fopen("/sys/bus/w1/devices/w1_bus_master1/
6             w1_master_slaves", "r");
7         if ($handle) {
8             while (($sensors = fgets($handle)) !== false) {
9                 $sensor = "/sys/bus/w1/devices/" . trim($sensors) . "/w1_slave";
10                $sensorhandle = fopen($sensor, "r");
11                if ($sensorhandle) {
12                    $thermometerReading = fread($sensorhandle, filesize($sensor)
13                );
14                    fclose($sensorhandle);
15                    preg_match("/t=(.+)/", preg_split("/\n/",
16                $thermometerReading)[1], $matches);
17                    $celsius = round($matches[1] / 1000, 3);
18
19                    $sensorsValue[] = sensor::create(trim($sensors), $celsius);
20                } else {
21                    throw new Exception("No sensor temperature found");
22                }
23            }
24            fclose($handle);
25        } else {
26            throw new Exception("No one wire found.");
27        }
28        return $sensorsValue;
29    }
30 }
```

Primer kode 5.2: Funkcija za branje vrednosti temperaturnih senzorjev.

## 5.5 Vklapljanje električnega grelca

Vklop in izklop električnega grelca je omogočen preko razreda gpio, ki je razširitev razreda awitch. Razred awitch skrbi za beleženje vseh vklopov in izklopov v bazo. Razred gpio pa za vklop ter izklop najprej GPIO priključka in s tem tudi električnega grelca. Ker programski jezik PHP nima knjižnice za krmljenje gpio priključkov, je to potrebno izvesti z izvajanjem zunanjih ukazov (primer kode 5.3).

```
1 class gpio extends awitch implements awitchInterface{
2
3     protected function switchOn(){
4         shell_exec("/usr/local/bin/gpio -g mode {$this->switch->gpio}
5         out");
6         shell_exec("/usr/local/bin/gpio -g write {$this->switch->gpio}
7         1");
8     }
9
10    protected function switchOff($lockRemove = false){
11        shell_exec("/usr/local/bin/gpio -g write {$this->switch->gpio}
12        0");
13    }
14
15    protected function switchGetCurrentStatus(){
16        return shell_exec("/usr/local/bin/gpio -g read {$this->switch->
17        gpio}");
18    }
19 }
```

Primer kode 5.3: Funkcija za vklop in izklop električnega grelca.



## Poglavje 6

# Optimizacija ogrevanja sanitarne vode

V tem poglavju ovrednotimo količino tople vode in jo izražamo v odstotkih. Prav tako ovrednotimo moč sončnega sevanja, zmogljivost električnega grelca in ocenimo porabo tople vode. Za analizo smo uporabili programsko okolje R [12]. Grafi so izrisani z R knjižnico ggplot2 [28].

Iz podatkovne baze smo zbrali pretekle zapise o sončnem sevanju, zunanje temperature in gibanje količine tople vode. S pomočjo računanja linearne regresije smo dobili funkcijo, ki prikazuje spremembo količine tople vode v odvisnosti od sončnega sevanja, zunanje temperature in ure v dnevu. Na podlagi pretekle rabe tople vode smo določili povprečno porabo tople vode glede na uro v dnevu.

### 6.1 Vrednotenje količine tople vode

Za potrebe delovanja in analize algoritma smo morali nekako ovrednotiti količino tople vode v bojlerju. Le tako lahko ocenimo, koliko tople vode porabimo, koliko jo je potrebno dogreti, ter kolikšna količina vode se bo segrela preko sončnih kolektorjev. Zaradi enostavnosti smo se odločili, da bomo prikazovali količino tople vode relativno, izraženo v odstotkih. S pomočjo testiranja smo ugotovili, da je  $35^{\circ}\text{C}$  najnižja sprejemljiva temperatura, primerna za uporabo. Zgornjo vrednost smo nastavili na  $85^{\circ}\text{C}$ , ki je zgornja temperatura, na katero naj bi segrevali bojler. Na samem bojlerju so štirje temperaturni senzorji, razmeščeni enakomerno od vrha do dna, zato smo razdelili bojler na tri toplotna območja, razmejena s toplotnimi senzorji. Za vsako temperaturno območje posebej izračunamo količino tople vode, pri čemer predpostavljamo,

da temperatura pada linearno od vrha proti dnu posameznega merilnega območja. Končna vrednost količine tople vode v bojlerju je povprečje vrednosti posameznih temperaturnih območij.

Predpostavimo, da imamo točki  $X1(0, S1)$  in  $X2(100, S2)$  v kartezičnem koordinatem sistemu, pri čemer sta vrednosti  $S1$  in  $S2$  temperaturi senzorjev  $S1$  in  $S2$ . Količino tople vode izračunamo kot površino pod daljico, ki jo določata točki  $X1$  in  $X2$ . Ker smo določili minimalno sprejemljivo temperaturo  $35^\circ\text{C}$ , se vzame samo površina kjer je  $y \geq 35$ . Izraz 6.1 prikazuje funkcijo za računanje površine posameznega temperaturnega območja.

$$P = \int_{start}^{konec} (kx + n)dx \quad (6.1)$$

Pri čemer so posamezne vrednosti:

- $k$  je koeficient in se izračuna po enačbi za izračun koeficienta linearne funkcije,
- $n$  je začetna višina, ki jo dobimo tako, da se od spodnjega senzorja ( $S1$ ) odšteje 35,
- vrednost start (spodnja meja) je določena kot minimalna vrednost  $x$  na grafu, kjer velja  $f(x) \geq 35$
- konec (zgornja meja) je določena z vrednostjo 100.

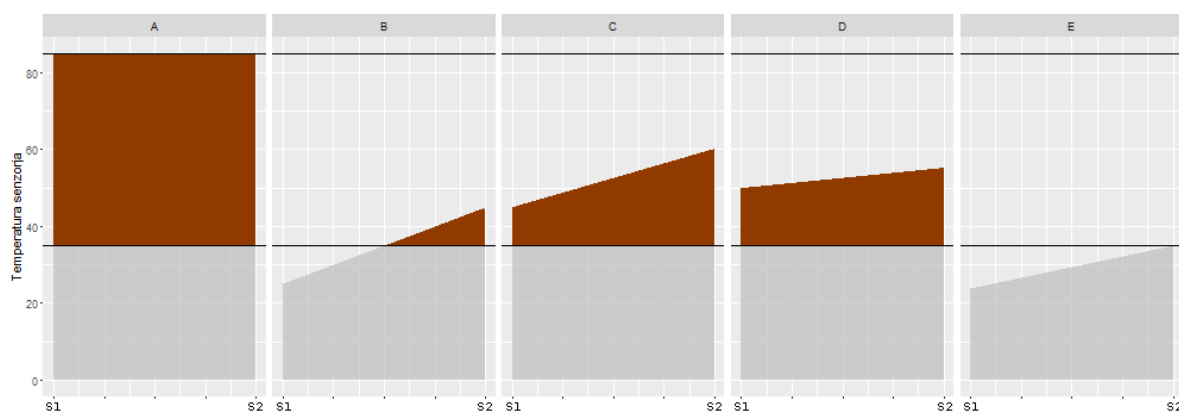
Vrednost količine tople vode posameznega območja smo določili z razmerjem izračunane površine količine vode glede na površino največje količine tople vode (senzorja  $S1$  in  $S2$  imata temperaturo  $85^\circ\text{C}$ ), izraženo v odstotkih.

Na sliki 6.1 prikazujemo ponazoritve količine tople vode v temperaturnem območju.  $S1$  označuje spodnji senzor v temperaturnem območju,  $S2$  pa zgornjega. S temno barvo na posameznem grafu prikazujemo količino tople vode.

Graf A na sliki 6.1 prikazuje največjo količino tople vode, ki ji določimo vrednost 100%. Temperatura senzorjev  $S1$  in  $S2$  je  $85^\circ\text{C}$ .

Graf B na sliki 6.1 prikazuje majhno količino tople vode. Temperatura na senzorju  $S1$  je  $25^\circ\text{C}$ , na senzorju  $S2$  pa  $45^\circ\text{C}$ . Razdalja med senzorjema na grafu je 100 enot. Izračunamo površino grafa, kjer je temperatura večja od  $35^\circ\text{C}$ . Površino delimo s površino grafa A in množimo s 100. Dobimo vrednost 5%, ki označuje razpoložljivo količino tople vode glede na največjo količino tople vode.

Pri grafu C na sliki 6.1 ima senzor  $S1$  vrednost  $45^\circ\text{C}$  senzor  $S2$  pa  $60^\circ\text{C}$ . Količina tople vode je 35%. V grafu D na sliki 6.1 pa ima senzor  $S1$   $50^\circ\text{C}$ ,



Slika 6.1: Ponazoritev količine tople vode v posameznem temperaturnem območju

senzor S2 pa 55 °C. Količina tople vode je prav tako 35%. Če primerjamo grafa C in D, vidimo da imata različne temperature na senzorjih, vendar je ocenjena količina tople vode enaka.

Graf D na sliki 6.1 prikazuje nezadostno količino tople vode. Vrednost je 0%.

Za potrebe računanja količine tople vode smo napisali funkcijo `departmentStorage`, ki sprejema vrednost senzorjev, ter vrača površino prikazano na sliki 6.1.

```

1 private static function departmentStorage($s1, $s2){
2     $s2 = $s2 - self::$minTemperature;
3     $s1 = $s1 - self::$minTemperature;
4
5     $k = ($s2 - $s1) / 100;
6     $n = $s1;
7
8     $start = 0;
9     $end = 100;
10
11     //removed scale - since we can have negative value / means it is
12     //way too low
13     if($s2 <= 0) return 0;
14     if($s1 <= 0) $start = -($n/$k);
15     if($n <= 0) $n = 0;
16
17     $volume = $k * pow(($end - $start), 2) / 2
18     + ($end - $start) * $n;
19
20     return $volume;
21 }

```

Primer kode 6.1: Funkcija za izračun zaloge tople vode v posameznem temperaturnem območju.

Končno stanje količine tople vode dobimo tako, da izračunamo povprečno vrednost temperaturnih območij. Primer izračuna je lepo viden v tabeli 6.1. Vrednosti S1 - S4 so vrednosti posameznih senzorjev, pri čemer je številčenje senzorjev od zgornje točke boilerja navzdol. Vrednost količine tople vode je izražena v procentih (stolpec "R v %").

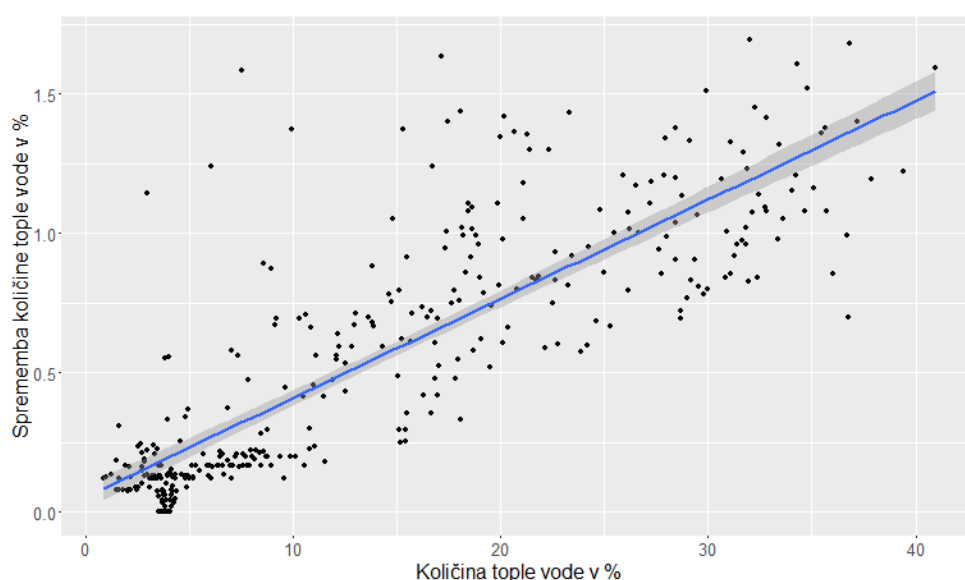
Meritev	S1	S2	S3	S4	R v %
1.	35.00 °C	35.00 °C	35.00 °C	35.00 °C	0.00
2.	37.00 °C	36.00 °C	35.01 °C	35.00 °C	1.34
3.	71.01 °C	63.25 °C	50.10 °C	46.25 °C	44.65
4.	71.01 °C	66.25 °C	50.10 °C	50.10 °C	47.93
5.	75.00 °C	69.00 °C	60.00 °C	55.00 °C	59.33
6.	85.00 °C	81.00 °C	80.00 °C	65.00 °C	87.33

Tabela 6.1: Primer izračuna količine tople vode glede na temperaturo senzorjev

## 6.2 Samoohlajanje boilerja

Med preliminarno raziskavo rabe tople vode smo opazili stalen počasen upad količine tople vode v časih, ko se ne uporablja. Ugotovili smo, da gre za samoohlajanje boilerja. To je posledica toplotnega toka iz višje temperature proti nižji, ki je večji v primeru slabše izolacije. Ker je bilo ohlajanje opazno, smo ga morali vključiti v svoj model.

Podatke za izračun vrednosti smo vzeli iz časov, ko se topla voda ne porablja, to je v času med polnočjo in tretjo uro zjutraj. Iz podatkov je bilo opazno, da je hitrost ohlajanja premosorazmerna z začetno količino tople vode. Rezultat merjenja smo prikazali na sliki 6.2. Točke na grafu prikazujejo posamezno



Slika 6.2: Graf praznjenja boilerja v odvisnosti od začetne količine tople vode

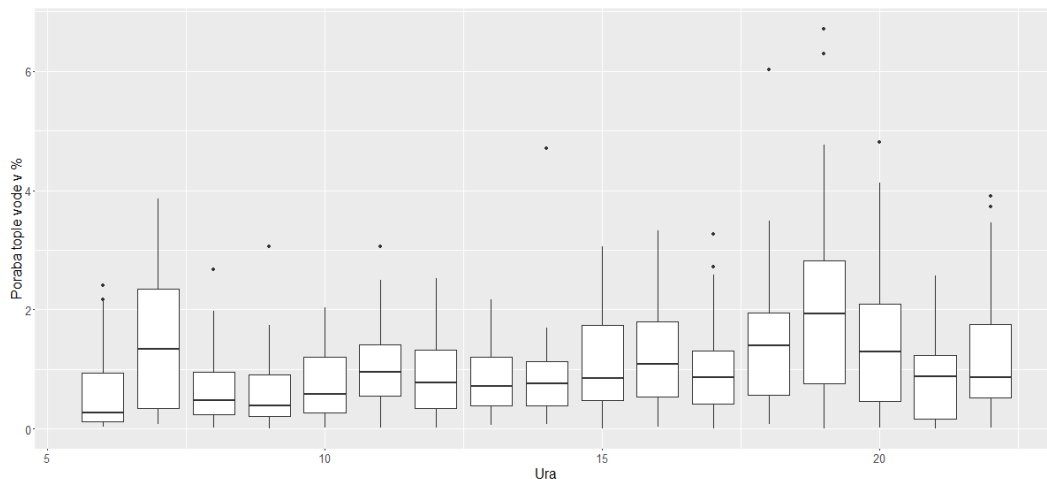
meritev. Vsaka točka prikazuje samoohlajanje boilerja v eni uri v odvisnosti od začetne količine tople vode. Začetna količina tople vode je vrednost, izmerjena ob začetku merjenja.

S pomočjo linearne regresije smo dobili funkcijo za izračun ohlajanja boilerja, pri čemer je  $f(x)$  ohladitev boilerja v 1 uri, izražena v odstotkih,  $x$  pa je začetna vrednost količine tople vode v opazovani uri (izraz 6.2). Ker je boiler postavljen v kleti, kjer je temperatura stalna in malo niha, nismo upoštevali še razlike med temperaturo boilerja in temperaturo zraka v kleti.

$$f(x) = 0,039 * x + 0,13 \quad (6.2)$$

## 6.3 Raba tople sanitarne vode

Podatke o rabi tople sanitarne vode potrebujemo za podporo odločanju, na kakšno vrednost moramo segrevati bojler v cenejši tarifi. Porabo tople vode smo dobili iz podatkov o zmanjšanju količine tople vode v opazovani uri. Odstranili smo znana skrajna primera porabe tople vode, kot sta polnjenje otroškega bazena in pranje avtomobila. Pri tem smo odšteli še praznjenje bojlerja zaradi samoohladitve. Podatki o rabi tople vode se stalno zbirajo ob obdobjih, ko je moč sončnega sevanja zanemarljiva. Na sliki 6.3 prikazujemo porabo vode od 6 ure zjutraj do 22 ure zvečer.



Slika 6.3: Graf porabe tople vode

Podatke o porabi tople vode smo zapisali v podatkovno bazo, od koder jih bere algoritem optimizacije.

## 6.4 Solarno ogrevanje bojlerja

Za meritve polnjenja bojlerja v odvisnosti od sončnega obsevanja smo vzeli intervale tekom dneva, ko ni bilo porabe tople vode. Sestavili smo tabelo meritev, na podlagi katere smo napisali funkcijo za izračun vpliva sončnega sevanja na polnjenje bojlerja.

Tabela 6.2 vsebuje informacije:

- čas, podan kot datum in ura v dnevu,

- napoved sončnega sevanja v  $\text{W/m}^2$  prikazuje skupno moč sončnega sevanja v podani uri (“neto\_radiation\_0day\_fc”),
- napoved povprečne temperature v  $^{\circ}\text{C}$  v podani uri (“temperature\_0day\_fc”),
- začetno stanje količine tople vode v odstotkih v pretekli uri (“full\_start”),
- sprememba količine tople vode v odstotkih v pretekli uri (“full\_change”).

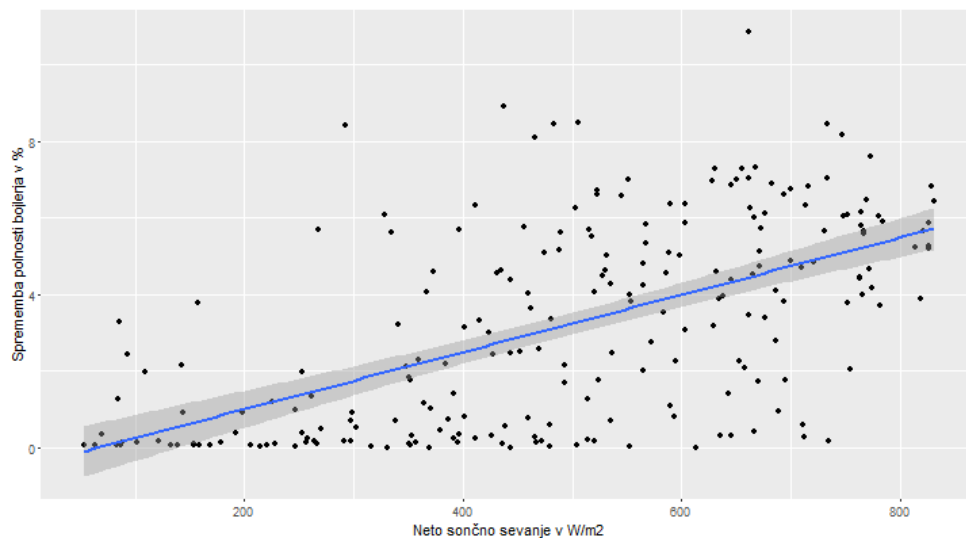
čas	napoved sončnega sevanja	napoved temperature	začetna količine tople vode	sprememba količine tople vode
	neto_radiation_0day_fc	temperature_0day_fc	full_start	full_change
2016-06-21 10	721 $\text{W/m}^2$	21.60 $^{\circ}\text{C}$	2,14%	4,845%
2016-06-21 14	634 $\text{W/m}^2$	24.00 $^{\circ}\text{C}$	31,18%	3,879%
2016-06-22 09	628 $\text{W/m}^2$	24.20 $^{\circ}\text{C}$	10,23%	6,959%
2016-05-15 09	291 $\text{W/m}^2$	11.10 $^{\circ}\text{C}$	3,79%	0,15%

Tabela 6.2: Primeri meritev sončnega obsevanja.

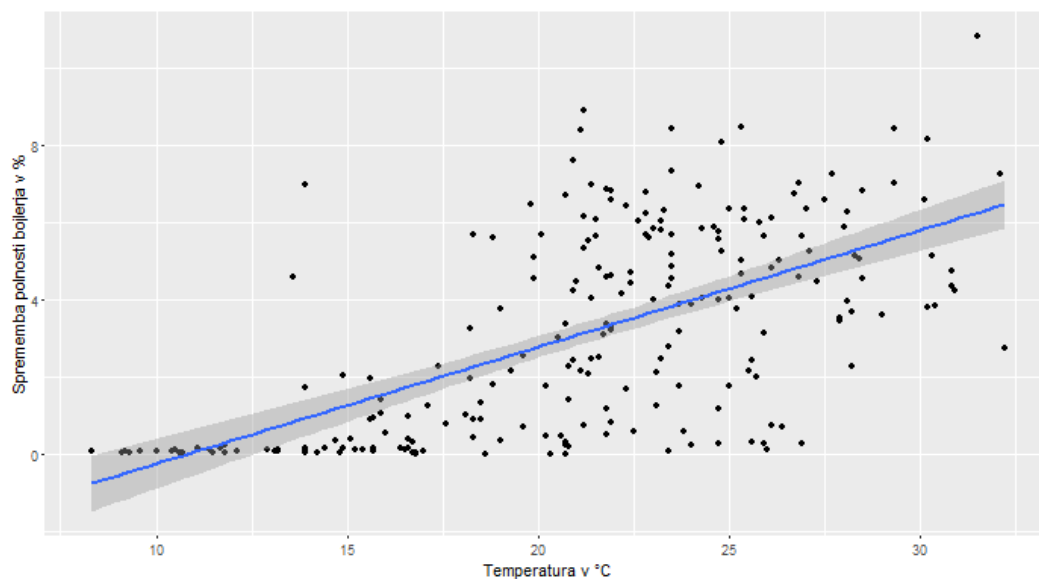
Na podlagi podatkov v tabeli smo izluščili model polnjenja bojlerja. Pomagali smo si z grafi, na katerih so prikazane spremembe količine tople vode na y osi v odvisnosti glede na opazovano znano vrednost. Vsaka pika na grafu prikazuje posamezno meritev.

Graf na sliki 6.4 prikazuje spremembo količine tople vode glede na predvideno sončno sevanje v opazovani uri. Odvisnost opazovanih spremenljivk je jasno vidna. Točke na dnu grafa so najverjetneje primeri napačne vremenske napovedi. Sevanje bi moralo biti, vendar se iz grafa vidi, da se količina tople vode ni spremenila. Sklepamo lahko, da je bilo v tistem trenutku oblačno.

Graf na sliki 6.5 prikazuje odvisnost spremembe količine tople vode glede na povprečno zunanjo temperaturo v opazovani uri. Odvisnost opazovanih spremenljivk je jasno vidna, kar je tudi posledica povezanosti povprečne temperature in sončnega sevanja. Točke na dnu grafa so meritve, ko ni bilo sončnega sevanja.



Slika 6.4: Graf spremembe količine tople vode glede na sončno sevanje

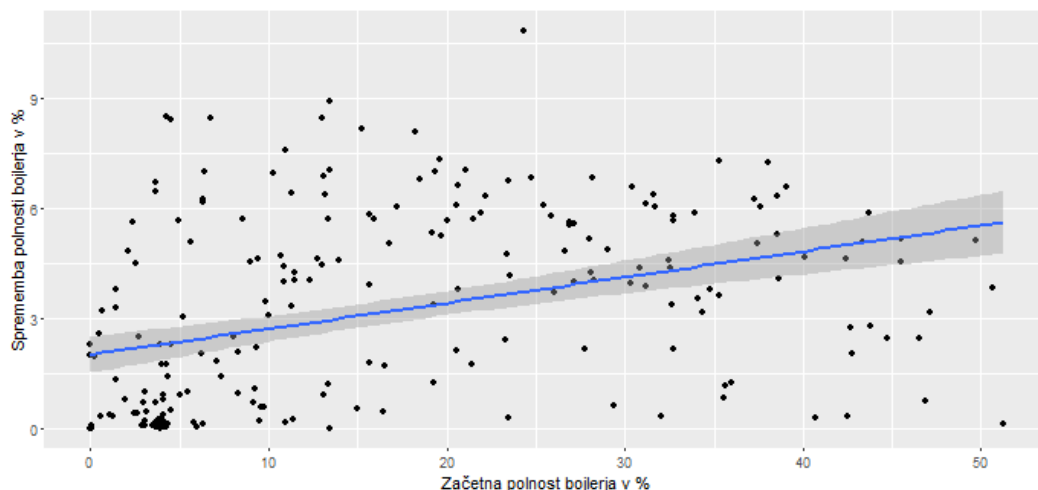


Slika 6.5: Graf spremembe količine tople vode glede na temperaturo

Graf na sliki 6.6 prikazuje spremembo količine tople vode glede na njeno začetno vrednost. Pričakovali smo odvisnost, da se boiler z večjo količino tople vode polni počasneje, vendar to iz grafa ni videti. Premica linearne regresije kaže na to, da sprememba količine tople vode v opazovani uri narašča



glede na začetno vrednost. To pa je v nasprotju z našimi izkušnjami, zato bomo previdni pri uporabi te spremenljivke v modelu. Graf prikazuje vrednosti začetne količine tople vode do okoli 50 odstotkov. V času meritev ni bilo primerov, ko bi bila količina tople vode v bojlerju večja.



Slika 6.6: Graf spremembe količine tople vode glede na začetno količino

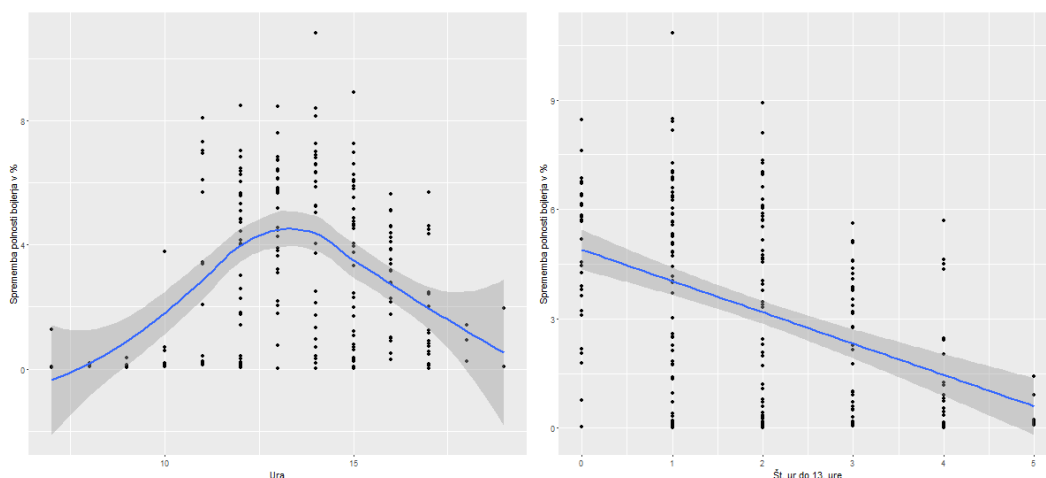
Ker sončni kolektorji niso idealno poravnani glede na pot sonca, nas je zanimala tudi odvisnost spremembe količine tople vode glede na uro v dnevu. Levi graf na sliki 6.7 prikazuje spremembo količine tople vode glede na uro v dnevu. Največje spremembe polnosti se pojavljajo okoli 13 ure, kar je nekako logično, saj je takrat tudi sončno obsevanje največje. Desni graf na sliki 6.7 pa prikazuje spremembo količine tople vode glede na to, za koliko ur je čas meritve oddaljen od 13 ure. Graf smo uporabili, ker lahko tukaj opišem odvisnost spremembe glede na uro s premico linearne regresije, medtem ko lahko odvisnost v levem grafu ponazorim le s polinomske regresijo.

Vse spremenljivke smo vnesli v funkcijo za računanje linearne regresije (primer kode 6.2), na podlagi katere smo dobili rezultate.

```
1 fit = lm( full_change ~ neto_radiation_0day_fc +
2   temperature_0day_fc + time_around_13 + full_start , tmp )
3 summary( fit )
```

Primer kode 6.2: Linearna regresija na izbranih spremenljivkah.

Tabela 6.3 prikazuje rezultat linearne regresije. Prvi stolpec vsebuje imena posameznih spremenljivk. Dodana je vrstica “(Intercept)”, ki ponazarja začetno vrednost v linearni funkciji. Stolpec “Estimate” v tabeli predstavlja smerni



Slika 6.7: Grafa spremembe količine tople vode glede na uro v dnevu

koeficient za posamezno spremenljivko. Stolpec “Std. error” predstavlja standardno napako koeficienta. Stolpec “Pr  $|t|$ ” prikazuje verjetnost, da spremenljivka ni pomembna. Manjša kot je vrednost, večji pomen ima spremenljivka. Če je vrednost “Pr  $|t|$ ” večja od 0.05, se predpostavlja, da je spremenljivka nepomembna. V tabeli 6.3 se vidi, da začetna količina tople vode (“full\_start”) po tem kriteriju ne vpliva na spremembo količine tople vode (“full\_change”). Popravili smo enačbo za linearno regresijo, tako da smo ta parameter odstranili. Ponovno smo zagnali računanje linearne regresije.

Coefficients:

	Estimate	Std. Error	t value	Pr $ t $	
(Intercept)	-2.1822202	0.8641403	-2.525	0.01226	*
neto_radiation_0day_fc	0.0029713	0.0009773	3.040	0.00265	**
temperature_0day_fc	0.2132088	0.0411351	5.183	4.92e-07	***
time_around_13	-0.2586818	0.1162030	-2.226	0.02702	*
full_start	-0.0064126	0.0141310	-0.454	0.65042	

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Tabela 6.3: Rezultat linearne regresije modela ogrevanja

Tabela 6.4 prikazuje popravljene rezultate linearne regresije. Če primerjamo stolpec “Pr  $|t|$ ” s stolpcem v tabeli 6.3, vidimo da so se vrednosti vseh

spremenljivk zmanjšale, kar je boljše. Z odstranitvijo nepomembne spremenljivke smo dobili boljši rezultat. Zato bomo uporabili popravljen izračun linearne regresije.

Coefficients:

	Estimate	Std. Error	t value	Pr >  t	
(Intercept)	-1.9533891	0.7004993	-2.789	0.00575	**
neto_radiation_0day_fc	0.0028867	0.0009577	3.014	0.00287	**
temperature_0day_fc	0.2013916	0.0317860	6.336	1.3e-09	***
time_around_13	-0.2788683	0.1071594	-2.602	0.00988	**

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

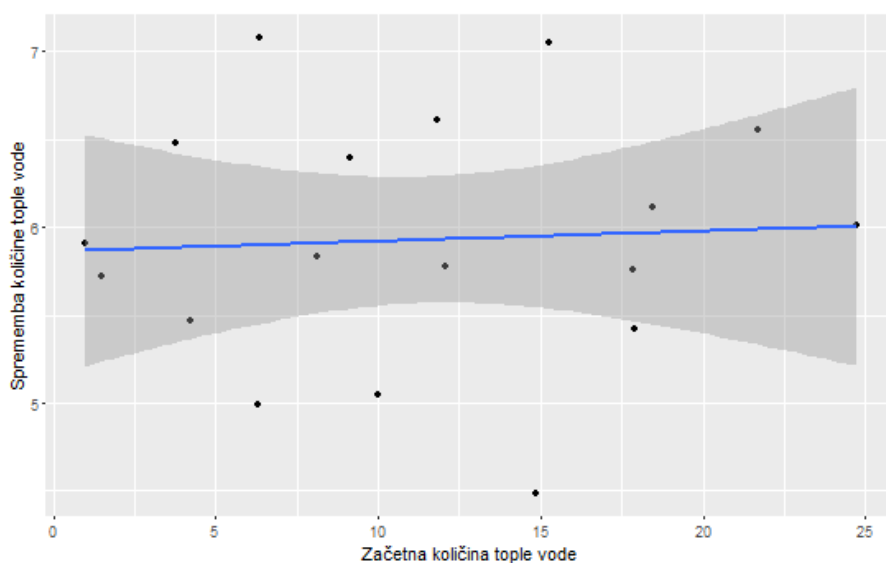
Tabela 6.4: Rezultat izboljšane linearne regresije modela ogrevanja

Na podlagi rezultata linearne regresije sestavimo funkcijo za računanje spremembe količine tople vode v posamezni uri. Polje "Estimate" v tabeli 6.4 predstavlja koeficient, s katerim moramo množiti vrednost posamezne spremenljivke. Funkcijo dobimo tako, da vsako spremenljivko zmnožimo z njenim koeficientom, nato vrednosti seštejemo in prištejemo še začetno vrednost (ki se v tabeli nahaja v vrstici "Intercept" stolpcu "Estimate").

$$f(x) = 0.00288 * neto\_radiation\_0day\_fc + 0.20139 * temperature\_0day\_fc - 0.27886 * time\_around\_13 - 1.95338 \quad (6.3)$$

## 6.5 Ogrevanje bojlerja z električnim grelcem

Za oceno časa, potrebnega za segretje bojlerja na določeno vrednost, potrebujemo oceno hitrosti ogrevanja s pomočjo električnega grelca. To smo naredili tako, da smo grelec vklapljali v določenih intervalih. Beležili smo spremembo količine tople vode ob vklopu in izklopu grelca. Na podlagi teh podatkov smo izračunali zmogljivost grelca. Na sliki 6.8 prikazujemo spremembo količine tople vode glede na začetno količino tople vode.



Slika 6.8: Graf delovanja električnega grelca v odvisnosti od začetne količine tople vode.

Iz zbranih podatkov vidimo, da je zmogljivost delovanja grelca rahlo odvisna od začetne količine tople vode. Vendar je ta odvisnost zanemarljiva in je nismo upoštevali.

Iz podatkov smo zato izračunali povprečno zmogljivost segrevanja bojlerja preko električnega grelca, ki znaša 5,9% na uro.

## 6.6 Algoritem optimizacije

Naloga algoritma za optimizacijo je zagotavljanje tople sanitarne vode z nižjimi stroški. To lahko dosežemo tako, da vodo ogrevamo na elektriko v cenejši tarifi in samo do potrebne količine. Algoritem optimizacije smo razdelili na dva algoritma.

Prvi algoritem se zaganja ponoči in skrbi za to, da se segreje zadostna količina vode za tekoči dan. Analizira vremensko napoved in preveri količino tople vode. Če je potrebno, vklopi grelec in segreje potrebno količino vode. V primeru, da za cel dan velja cenejša cena električne energije, algoritem ponoči ne vklopi grelca.

Drugi algoritem stalno preverja, ali je na voljo minimalna količina tople vode. Grelec vkloplja po potrebi. Služi kot rezerva v primeru, da je prvi algoritem segrel premajhno količino vode. V sobotah, nedeljah ter ob praznikih, ko je cena električne energije nižja skozi ves dan, pa prevzame glavno vlogo zagotavljanja tople vode.

### 6.6.1 Algoritem solarne optimizacije

Algoritem optimizacije solarnega ogrevanja se izvršuje med polnočjo in 5 uro zjutraj na vsako polno uro. Za to skrbi “crontab” v Linuxu. Razlog za periodično izvrševanje je občasna zamuda pri pošiljanju vremenske napovedi iz ARSO in zadnja napoved ni vedno pravočasna. Shema na sliki 6.9 prikazuje poenostavljeno delovanje algoritma.

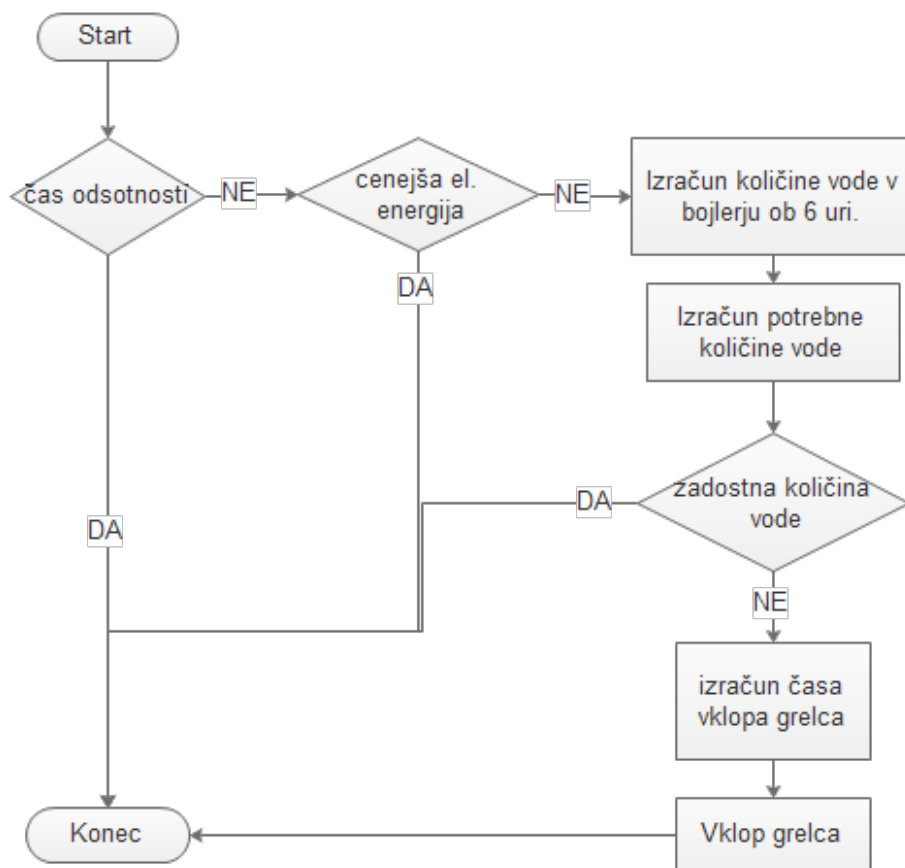
Algoritem najprej preveri, ali smo vnesli čas odsotnosti. V primeru, da smo odsotni, se zaključi. Preveri tudi, ali je cena električne energije cenejša skozi cel dan in se v tem primeru zaključi. Drugače pa algoritem izračuna količino tople vode, ki bo na voljo ob 6 uri zjutraj. Pri tem upošteva samoohlajanje bojlerja. Nato izračuna minimalno količino tople vode, ki jo potrebujemo zjutraj glede na vremensko napoved.

V primeru, da je predvidena količina tople vode nižja od potrebne, izračuna potrebni čas vklopa grelca. Grelec vklopi in zaklene vklop za predviden čas. Uporabili smo metodo zaklepa, ki prepreči izklop grelca v podanem obdobju. Metoda je praktična, ker hkrati stalno teče tudi algoritem zagotavljanja minimalne količine tople vode, ki bi grelec izklopil. Algoritem se nato zaključi. Za izklop grelca pa poskrbi algoritem za zagotavljanje minimalne količine tople vode.

Primer kode 6.3 prikazuje funkcijo za izračun potrebne minimalne količine vode. Uporablja metodo simulacije, pri čemer se v zanki sprehodi skozi vsako posamezno uro in izračuna potrebo po dogretju vode. Potrebo po porabi vode računa v obratni smeri urinega kazalca od 22 ure zvečer proti 6 uri zjutraj. Tak način smo izbrali, ker enostavneje upoštevamo samoohlajanje bojlerja.

Za vsako opazovano uro izračuna porabo tople vode, količino ogrete vode s sončnimi kolektorji ter samoohlajanje bojlerja. Na podlagi tega dobi minimalno potrebno količino tople vode, ki jo prenese v naslednjo ponovitev. Ker

zaloge tople vode ne moremo prenesti nazaj, se v primeru negativne vrednosti potrebe tople vode dodeli vrednost 0% (primer kode 6.3).



Slika 6.9: Poenostavljen diagram poteka algoritma optimizacije

```

1  /**
2   * Calculate required boiler warm at 6
3   *
4   * @return percentage of boiler full
5   */
6  public static function SolarDIALhghorithm($time = "now"){
7      $time = electricSupply::getTime($time);
8
9      $prediction = [];
10     $prevPred = new prediction();
11
12     for($i = 22; $i >= 6; $i--){
13         $pred = new prediction();
14         $pred->hour = $i;
15         $pred->time = strtotime(date('Y-m-d $i:00:00', $time));
16         $pred->consumption = self::getHotWaterConsumption($i);
17         $pred->cooling = self::fullAfterHours($pred->consumption +
18             $prevPred->minBoilerFull, 1);
19         $pred->sunHeat = self::getSunHeat($pred->time, $pred->
20             minBoilerFull);
21
22         $pred->minBoilerFull = max($pred->consumption + $prevPred->
23             minBoilerFull + $pred->cooling - $pred->sunHeat, 0);
24
25         $prediction[] = $pred;
26         $prevPred = $pred;
27     }
28
29     return $pred->minBoilerFull;
30 }

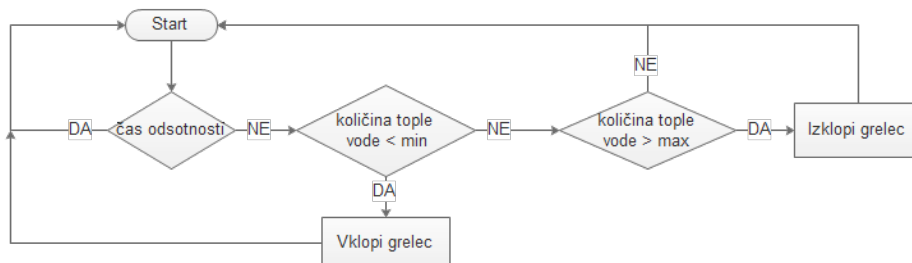
```

Primer kode 6.3: Izračun potrebne količine tople vode

### 6.6.2 Algoritem za zagotavljanje minimalne količine tople vode

Poenostavljen diagram poteka na sliki 6.10 prikazuje delovanje algoritma. Algoritem teče v neskončni zanki in stalno preverja količino tople vode. Če količina tople vode pade pod določeno minimalno vrednost, vklopi grelec. Ko količina vode naraste nad določeno najvišjo vrednost, pa grelec izklopi. Izjema je, kadar je vklop grelca zaklenjen (zaklep po potrebi nastavi algoritem solarne optimizacije). Takrat mora algoritem počakati, da se zaklep sprostí. V primerih, ko smo odsotni, pa se grelec ne vkloplja.

Primer kode 6.4 prikazuje implementacijo algoritma.



Slika 6.10: Poenostavljen diagram poteka algoritma zagotavljanja minimalne količine vode

```

1 public static function allDayCheapCurrentAlgorithm() {
2   self::init();
3   $switch = awitch::get('boiler');
4   while(true) {
5     $time = time();
6     $boiler = get::getBoilerData();
7     if(!holidayTimes::areWeOnHoliday() && round($boiler->status_full
8       ,2) <= self::$minBoilerFull) {
9       if($switch->getCurrentStatus() == 0) {
10        $switch->on();
11      }
12    }
13    elseif($boiler->status_full > self::$maxBoilerFull ||
14      holidayTimes::areWeOnHoliday()) {
15      if($switch->getCurrentStatus() == 1) {
16        $switch->off();
17      }
18    }
19    sleep(60 - (time() - $time));
20  }
21 }

```

Primer kode 6.4: Algoritem za zagotavljanje minimalne količine tople vode.



## Poglavje 7

# Delovanje sistema

V tem poglavju smo se osredotočili predvsem na učinkovitost in smotrnost delovanja sistema. Ocenili bomo prihranek optimizacije vklapljanja grelca ob določenih posplošenih vremenskih pogojih glede na prejšnji sistem, ko se je grelec vklapljal po potrebi. Prav tako bomo na hitro ocenili možne izboljšave in njihovo smotrnost. Oceno stroškov bomo naredili s pomočjo simulacije, ki jo bomo izvajali s pomočjo zgrajenih modelov za posamezne procese. Te smo oblikovali na osnovi meritev v realnem testnem okolju. Simulirali bomo dneve z različnimi vremenskimi pogoji, pri čemer predpostavljamo, da je vremenska napoved točna. Nato pa bomo na različnih vremenskih pogojih simulirali delovanje starega in novega sistema.

Simulirali bomo naslednje tipične primere dnevov:

- sončen dan - sonce na polno sije preko celega dneva,
- jutranja oblačnost - dopoldne imamo oblačnost, popoldne sije sonce,
- popoldanska oblačnost - dopoldne sije sonce, popoldne je oblačno,
- delna oblačnost - sonce je občasno prekrito z oblaki preko celega dneva, sončno sevanje je okoli 30% manjše kot na sončen dan,
- pretežna oblačnost - sonce je pretežno zakrito z oblaki preko celega dneva, sončno sevanje je okoli 60% manjše kot na sončen dan,
- oblačen dan - sončno sevanje je majhno, več kot 80% manjše kot na sončen dan,
- zelo oblačen dan - sončno sevanje je več kot 90% manjše kot na sončen dan.

V simulaciji se predpostavi, da je količina tople vode ob 6. uri zjutraj enaka 0%. Za kriterij učinkovitosti in primerjavo bomo uporabljali strošek električne energije. Algoritem, ki ob enakih pogojih ceneje zagotavlja potrebno količino tople vode je boljši.

## 7.1 Simulacija A: obstoječe samoohlajanje bojlerja

V tej simulaciji primerjamo strošek ogrevanja sanitarne vode med starim sistemom, ki je vodo ogreval po potrebi, in novim sistemom, ki vso potrebno vodo segreje v intervalu cenejše električne energije. Vse vrednosti testiramo na iste dneve pod enakimi pogoji.

Podatki rezultatov so zbrani v tabeli 7.1. V tabeli prikazujemo podatke o porabi električne energije (stolpec “por. v kWh”) ter strošek električne energije (stolpec “str. v €”) med enim in drugim načinom. V stolpcu “%” prikazujemo, za koliko odstotkov je strošek ogrevanja vode z algoritmom solarne optimizacije večji ali manjši glede na strošek ogrevanja po potrebi. Kadar je odstotek večji od nič, govorimo o prihranku, drugače pa o izgubi.

Iz tabele se lepo vidi, da je poraba električne energije vedno večja v primeru ogrevanja tople vode samo v cenejši tarifi. To je nekako logično, saj je potrebno upoštevati še izgube toplote v samem bojlerju (samoohlajanje).

Vremenska situacija	Ogr. po potrebi		Opt. ogrevanja		%
	por. v kWh	str. v €	por. v kWh	str. v €	
sončen dan	0,70kWh	0,09€	0,75kWh	0,06€	+33%
jutranja oblačnost	1,38kWh	0,17€	1,47kWh	0,12€	+29%
popoldanska oblačnost	1,82kWh	0,19€	2,99kWh	0,25€	-31%
delna oblačnost	1,23kWh	0,15€	1,55kWh	0,13€	+13%
pretežna oblačnost	4,96kWh	0,60€	8,18kWh	0,69€	-15%
oblačen dan	8,56kWh	1,06€	13,66kWh	1,16€	-9%
zelo oblačen dan	11,04kWh	1,36€	16,76kWh	1,41€	-3%

Tabela 7.1: Primerjava učinkovitosti ogrevanja po potrebi in optimizacije ogrevanja.

Pričakovali bi, da bo gretje vode samo v nizki ceni električne energije cenejše, vendar lahko iz primerjave stroškov ogrevanja vidimo, da ni vedno tako. V primerih popoldanske oblačnosti, pretežne oblačnosti, oblačnega ter zelo oblačnega dneva vidimo, da bi dejansko z gretjem tople vode v cenejšem toku za ogrevanje plačali več, kot bi plačali, če bi toplo vodo ogrevali samo po potrebi. Razlog za to pa je predvsem prevelika izguba toplote zaradi slabe izolacije na bojlerju. Zaradi samoohlajanja bojlerja je potrebno segreti mnogo več tople vode, ki se pa kljub temu, da jo grejemo v cenejši električni tarifi, ne splača.

Ugotovitev v tem primeru je, da uporaba algoritma v tem konkretnem primeru testnega okolja ni smiselna. Vprašamo se, kaj bi se zgodilo, če bi lahko samoohlajanje bojlerja upočasnili oziroma zmanjšali z dodatno izolacijo?

## 7.2 Simulacija B: zmanjšano samoohlajanje bojlerja

V prejšnji simulaciji smo ugotovili, da ogrevanje tople vode samo v cenejši tarifi ni smiselno. Zato smo se odločili simulirati enak algoritem pod drugimi pogoji. Bojler je namreč slabo izoliran. Cevi, ki so priklopljene na bojler niso izolirane. Postavitev bojlerja v konkretnem primeru omogoča, da ga dodatno izoliramo. Menimo, da bi lahko toplotne izgube zmanjšali za polovico, zato bomo to simulacijo izvajali ob predpostavki, da je samoohlajanje bojlerja manjše za polovico. Tabela 7.2 prikazuje učinkovitost delovanja algoritma.

Iz tabele se lepo vidi, da se ob boljši izolaciji bojlerja tehnica prevesi k optimizaciji ogrevanja. Ker so toplotne izgube sedaj mnogo manjše, je smiselno ogrevati vodo v času nizke cene električne energije. Sedaj je algoritem boljši v vseh vremenskih situacijah, razen v simulaciji oblačnega dneva. Po analizi smo ugotovili, da ima ta dan večjo oblačnost v popoldanskem času kot dopoldanskem, zato je prišlo do slabšega rezultata. Ocenjujemo, da je uporaba algoritma za ogrevanje smotrna v primeru izboljšane izolacije bojlerja in priključnih cevi.

## 7.3 Simulacija C: izboljšan regulacijski algoritem

V tej simulaciji smo poskušali še izboljšati delovanje sistema ob nespremenjenih pogojih. Odločili smo se za popravek algoritma. Algoritem bo sedaj

Vremenska situacija	Ogr. po potrebi		Opt. ogrevanja		%
	por. v kWh	str. v €	por. v kWh	str. v €	
sončen dan	0,67kWh	0,09€	0,69kWh	0,06€	+33%
jutranja oblačnost	1,31kWh	0,17€	1,35kWh	0,11€	+35%
popoldanska oblačnost	0,57kWh	0,07€	0,59kWh	0,05€	+28%
delna oblačnost	0,97kWh	0,12€	0,99kWh	0,10€	+16%
pretežna oblačnost	3,98kWh	0,47€	4,91kWh	0,42€	+10%
oblačen dan	7,67kWh	0,98€	9,78kWh	0,99€	-1%
zelo oblačen dan	10,47kWh	1,30€	12,55kWh	1,06€	+18%

Tabela 7.2: Primerjava učinkovitosti ogrevanja po potrebi in optimizacije ogrevanja v primeru izboljšane izolacije.

predvideval samo segrevanje vode do 13. ure in bo v času cenejše električne energije segrel samo potrebno količino vode, porabljene do 13. ure. Po 13. uri bo nato toplo vodo segreval po potrebi.

Rezultati so prikazani v tabeli 7.3. Iz rezultatov se vidi, da je v tem primeru delovanje algoritma smotno v vseh opazovanih situacijah, je pa res razlika sedaj manjša, kot je razlika v primeru, da imamo izboljšano izolacijo. Ker bojlerja nismo dodatno izolirali, se je ta algoritem izkazal za najbolj učinkovitega in je trenutno uporabljen v testnem okolju.

## 7.4 Ocena učinkovitosti izboljšanega regulacijskega algoritma

Za oceno učinkovitosti izboljšanega regulacijskega algoritma skozi daljše obdobje, potrebujemo število sončnih in oblačnih dni tekom leta. Iz mesečnega biltena Naše okolje [29], ki ga izdaja ARSO, smo vzeli podatke o številu sončnih dni, številu oblačnih dni, ter povprečno oblačnost za leto 2015. Podatke posameznih biltenov smo zbrali v tabeli 7.4. Tabela vsebuje podatke za mesece, v katerih uporabljamo električni grelec.

Podatke smo nato povezali s stroški ogrevanja po potrebi in stroški ogrevanja izboljšanega regulacijskega algoritma (tabela 7.3). Število jasnih dni smo množili s stroškom ogrevanja z vremensko situacijo sončnega dneva. Število oblačnih dni smo množili s stroškom ogrevanja za oblačen dan. Za preostale

Vremenska situacija	Ogr. po potrebi		Opt. ogrevanja		%
	por. v kWh	str. v €	por. v kWh	str. v €	
sončen dan	0,70kWh	0,09€	0,75kWh	0,06€	+33%
jutranja oblačnost	1,38kWh	0,17€	1,47kWh	0,12€	+29%
popoldanska oblačnost	1,82kWh	0,19€	1.86kWh	0,16€	+15%
delna oblačnost	1,23kWh	0,15€	1,30kWh	0,13€	+13%
pretežna oblačnost	4,96kWh	0,60€	5,45kWh	0,54€	+10%
oblačen dan	8,56kWh	1,06€	8.73kWh	0.98€	+7%
zelo oblačen dan	11,04kWh	1,36€	11,32kWh	1,27€	+6%

Tabela 7.3: Primerjava učinkovitosti ogrevanja po potrebi in popravljenega algoritma optimizacije ogrevanja

dneve v mesecu pa smo določili vremensko situacijo glede na povprečno oblačnost (v desetinah). Kadar je povprečna oblačnost večja od 6 smo upoštevali strošek ogrevanja za pretežno oblačen dan. V ostalih primerih pa smo upoštevali strošek ogrevanja za delno oblačen dan.

V tabeli 7.5 prikazujemo ocenjeno učinkovitost algoritma optimizacije. Za vsak mesec prikažemo strošek ogrevanja po potrebi in strošek ogrevanja z uporabo izboljšanega regulacijskega algoritma. Podamo razliko v evrih in odstotkih. Na koncu prikažemo skupen strošek ogrevanja. Iz tabele lahko povzamemo, da izboljšan regulacijski algoritem zagotavlja toplo vodo približno 9% ceneje, kot ogrevanje vode po potrebi.

mesec	št. jasnih dni	št. oblačnih dni	povprečna oblačnost (v desetinah)
marec	6	7	5
april	5	4	4.6
maj	1	9	6.2
junij	3	4	5.1
julij	5	5	4.2
avgust	7	7	4.7
september	2	10	6.5
oktober	0	14	7.6
november	3	7	5.6

Tabela 7.4: Število jasnih dni, število oblačnih dni ter povprečna oblačnost po mesecih v letu 2015

mesec	Ogr. po potrebi str. v €	Opt. ogrevanja str. v €	Razlika	
			v €	v %
marec	10.66€	9.56€	1.1€	10.32%
april	7.84€	6.95€	0.89€	11.35%
maj	22.3€	20.22€	2.01€	9.04%
junij	7.96€	7.09€	0.87€	10.93%
julij	8.9€	7.93€	0.97€	10.90%
avgust	10.6€	9.49€	1.11€	10.47%
september	21.58€	19.64€	1.94€	8.99%
oktober	25.04€	22.90€	2.14€	8.55%
november	19.69€	17.84€	1.85€	9.40%
skupaj	134.5€	121.62€	12.88 €	9.58%

Tabela 7.5: Primerjava ocenjenih stroškov ogrevanja po potrebi in optimiziranega ogrevanja po posameznih mesecih

## Poglavje 8

### Sklepne ugotovitve

V diplomski nalogi smo želeli izboljšati ogrevanje sanitarne vode. Sestavili smo regulator, ki meri količino tople vode v bojlerju. Preko elektronske pošte dobiva vremensko napoved, v kateri so informacije o sončnem sevanju. Na podlagi danih vrednosti smo zgradili model ogrevanja sanitarne vode, ki segreje toliko tople vode, kot je potrebno. Preko spletnega vmesnika lahko nadzorujemo delovanje sistema in vnašamo čase odsotnosti, ko se grelec popolnoma izklopi.

Ugotovili pa smo tudi pomen dobre izolacije na bojlerju in prikazali, da se gretje tople vode vnaprej ne splača vedno. V testnem sistemu se je pokazalo, da je izguba tople vode na podlagi ohlajanja preko sten bojlerja enostavno prevelika, da bi pokrila cenejše ogrevanje.

Ocenjujemo, da je sam model optimizacije ustrezen. Večje število meritev bi pripomoglo k bolj točni napovedi potrebne tople sanitarne vode. Razmisliti pa bi morali tudi, če bi se splačalo v model vključiti še trenutni mesec v letu. Ker se kot, pod katerim sije sonce na zemljo, spreminja tekom leta, ima vpliv na delovanje sončnih kolektorjev. Zaradi premajhnega števila zbranih meritev ni bil upoštevan v diplomu.

Kartični računalnik je za potrebe same optimizacije premočan, zato bi bilo mogoče dobro sam model vključiti v kakšen drug sistem pametne hiše, ali pa uporabiti kakšen manj zmogljiv, manj potraten in cenejši kartični računalnik kot je Raspberry Pi Zero, ali pa celo kakšen mikrokontroler Arduino Mega.

Prihranki pri ceni električne energije pokrivajo delovanje rešitve, vendar pa so zaradi relativno majhne porabe majhni. Zanimivo bi bilo implementirati rešitev na večjih sistemih. Razmisliti pa bi bilo dobro tudi, če bi lahko isti model uporabili tudi na drugih sistemih kot so npr. banke ledu [3,6], pasivno ogrevanje, itd...

# Slike

2.1	Shema trenutnega sistema ogrevanja sanitarne vode . . . . .	5
4.1	Shema povezav strojne opreme regulatorja . . . . .	9
4.2	Raspberry Pi 2 model B. . . . .	10
4.3	Raspberry Pi 2 seznam priključkov . . . . .	12
4.4	Temperaturni senzor DS18B20 . . . . .	12
4.5	Razdelilna letev za temperaturne senzorje . . . . .	13
4.6	Shema vezja za vklop grelca . . . . .	14
4.7	Izvedba vezja za vklop grelca . . . . .	14
4.8	Optosklopnik 4N35 . . . . .	15
4.9	Tranzistor BC457 . . . . .	16
4.10	Rele JW2SN . . . . .	16
4.11	Kontaktor IKA20-20 . . . . .	17
4.12	Sestavljen regulator ogrevanja . . . . .	18
5.1	Shema programske opreme . . . . .	20
5.2	Uporabniški vmesnik . . . . .	21
5.3	Graf gibanja količine tople vode tekom dneva . . . . .	22
5.4	Prikaz temperaturnih senzorjev v operacijskem sistemu . . . . .	24
5.5	Prikaz zapisa vrednosti temperaturnega senzorja . . . . .	25
6.1	Ponazoritev količine tople vode v posameznem temperaturnem območju . . . . .	29
6.2	Graf praznjenja bojlerja v odvisnosti od začetne količine tople vode . . . . .	31
6.3	Graf porabe tople vode . . . . .	32
6.4	Graf spremembe količine tople vode glede na sončno sevanje . . . . .	34
6.5	Graf spremembe količine tople vode glede na temperaturo . . . . .	34
6.6	Graf spremembe količine tople vode glede na začetno količino . . . . .	35
6.7	Grafa spremembe količine tople vode glede na uro v dnevu . . . . .	36



6.8	Graf delovanja električnega grelca v odvisnosti od začetne količine tople vode . . . . .	38
6.9	Poenostavljen diagram poteka algoritma optimizacije . . . . .	40
6.10	Poenostavljen diagram poteka algoritma zagotavljanja minimalne količine vode . . . . .	42

# Tabele

6.1	Primer izračuna količine tople vode glede na temperaturo senzorjev . . . . .	30
6.2	Primeri meritev sončnega obsevanja. . . . .	33
6.3	Rezultat linearne regresije modela ogrevanja . . . . .	36
6.4	Rezultat izboljšane linearne regresije modela ogrevanja . . . . .	37
7.1	Primerjava učinkovitosti ogrevanja po potrebi in optimiziranega ogrevanja. . . . .	44
7.2	Primerjava učinkovitosti ogrevanja po potrebi in optimizacije ogrevanja v primeru izboljšane izolacije. . . . .	46
7.3	Primerjava učinkovitosti ogrevanja po potrebi in popravljenega algoritma optimizacije ogrevanja . . . . .	47
7.4	Število jasnih dni, število oblačnih dni ter povprečna oblačnost po mesecih v letu 2015 . . . . .	48
7.5	Primerjava ocenjenih stroškov ogrevanja po potrebi in optimiziranega ogrevanja po posameznih mesecih . . . . .	48

# Seznam primerov kode

5.1	Primer vremenske napovedi v XML obliki . . . . .	24
5.2	Funkcija za branje vrednosti temperaturnih senzorjev. . . . .	25
5.3	Funkcija za vklop in izklop električnega grelca. . . . .	26
6.1	Funkcija za izračun zaloge tople vode v posameznem temperaturnem območju. . . . .	30
6.2	Linearna regresija na izbranih spremenljivkah. . . . .	35
6.3	Izračun potrebne količine tople vode . . . . .	41
6.4	Algoritem za zagotavljanje minimalne količine tople vode. . . . .	42

# Literatura

- [1] aliimg.com. Slika BC 457. Dostopno na [http://i00.i.aliimg.com/img/pb/611/486/942/942486611\\_849.jpg](http://i00.i.aliimg.com/img/pb/611/486/942/942486611_849.jpg).
- [2] Branislav Šarčević. Shema trenutnega ogrevanja sanitarne vode. Dostopno na <https://icq.com/files/0jcjcQlaiB5fGcc9pYsyCy573efbff1be>.
- [3] Častnik Finance d.o.o. Banka ledu bo hladila ljubljanski grad. Dostopno na <http://oe.finance.si/8843963>.
- [4] Neznani avtor. Slika optosklopnika 4N35. Dostopno na <https://lh4.googleusercontent.com/-i2jNpVfu4pE/TXCqqqAlxvI/AAAAAAAAAZ4/C0zZJLcWymI/s1600/4N35+opto.png>.
- [5] Oracle Corporation. MySQL podatkovna baza. Dostopno na <http://www.mysql.com>.
- [6] Emicon Adria d.o.o. Banke ledu. Dostopno na [http://www.bankaledu.si/banke\\_ledu.html](http://www.bankaledu.si/banke_ledu.html).
- [7] Energetika.NET. Uporaba sončnih kolektorjev. Dostopno na <http://www.energijadoma.si/novice/uporaba-soncnih-kolektorjev>.
- [8] Fairchild. Tranzistor BC457. Dostopno na <https://www.fairchildsemi.com/datasheets/BC/BC547.pdf>.
- [9] Farnell. Slika releja JW2SN. Dostopno na [https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcSfA0Xf14My20NwCsoKX\\_FJJK3CYHPKrGB2\\_KixZR3aODkRzP4iCw](https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcSfA0Xf14My20NwCsoKX_FJJK3CYHPKrGB2_KixZR3aODkRzP4iCw).
- [10] Apache Software Foundation. Apache 2 spletni strežnik. Dostopno na <http://www.apache.org>.

- [11] Raspberry Pi Foundation. Raspberry Pi 2 model B specifikacija. Dostopno na <https://www.raspberrypi.org/products/raspberry-pi-2-model-b>.
- [12] The R Foundation. Programsko okolje R. Dostopno na <https://www.r-project.org>.
- [13] PHP group. Programski jezik PHP. Dostopno na <http://php.net>.
- [14] Highsoft. Grafi Highcharts. Dostopno na <http://www.highcharts.com>.
- [15] Mark Otto in Jacob. Bootstrap ogrodje. Dostopno na <http://getbootstrap.com/2.3.2>.
- [16] Martin Wimpress in Rohith Madhavan. Ubuntu Mate. Dostopno na <https://ubuntu-mate.org/raspberry-pi>.
- [17] Texas Instruments. 4N35 Optosklopnik. Dostopno na <http://www.ti.com/lit/ds/symlink/4n35.pdf>.
- [18] Maxim Integrated. DS18B20 digital thermometer. Dostopno na <https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html>.
- [19] Elektro in elektronska industrija d.d. Iskra. Kontaktor IKA20 specifikacija. Dostopno na <http://www.iskra.eu/mma/IKA20-UL%20IKA63-UL/2012021307392315>.
- [20] Yii Software LLC. Yii2. Dostopno na <http://www.yiiframework.com>.
- [21] ARM Ltd. Cortex A7 Processor. Dostopno na <http://www.arm.com/products/processors/cortex-a/cortex-a7.php>.
- [22] Panasonic. Rele JW2SN-5VDC. Dostopno na [https://www3.panasonic.biz/ac/e\\_download/control/relay/power/catalog/mech\\_eng\\_jw.pdf](https://www3.panasonic.biz/ac/e_download/control/relay/power/catalog/mech_eng_jw.pdf).
- [23] PrivateEyePi. DS18B20 temperaturni senzor slika. Dostopno na <http://ha.privateeyepi.com/store/image/cache/data/tempsensor-500x500.jpg>.
- [24] Raspberrypi.org. Slika Raspberry PI. Dostopno na <https://www.raspberrypi.org/wp-content/uploads/2015/01/Pi2ModB1GB-comp.jpeg>.

- [25] Redis.io. Redis podatkovni medpomnilnik. Dostopno na <http://redis.io>.
- [26] SwampieUK. Slika GPIO Raspberry Pi. Dostopno na <https://www.raspberrypi.org/forums/viewtopic.php?t=112363&p=771032>.
- [27] JoKe Systems. Kontaktor IKA20-20. Dostopno na [http://b2c.joke-systems.at/product\\_info.php?info=p2667\\_IKA20-20-24.html](http://b2c.joke-systems.at/product_info.php?info=p2667_IKA20-20-24.html).
- [28] Hadley Wickham. Knjižnica za risanje grafov v okolju R ggplot2. Dostopno na <http://ggplot2.org>.
- [29] Agencija Republike Slovenije za okolje. Mesečni bilten ARSO - Letnik 2015. Dostopno na <http://www.arso.gov.si/o%20agenciji/knji%C5%BEnica/mese%C4%8Dni%20bilten/bilten2015.htm>.
- [30] Agencija za učinkovito rabo energije. Priprava tople sanitarne vode. Dostopno na <http://gcs.gi-zrmk.si/Svetovanje/Clanki/PDFknjiznjicaAURE/IL1-16.PDF>.